AD-A272 280

(2)

SC71057FR

SC71057FR

Copy No. 3

SPREAD SPECTRUM RANDOM ACCESS NETWORKS

FINAL REPORT FOR THE PERIOD
September 1, 1991 through August 31, 1993

Contract No. N00014-91-C-0234

Prepared for

BMDO/Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Attn: Dr. Rabinder N. Madan

Ambatipudi R. Sastry
Principal Investigator

October 1993

Rockwell International
Science Center

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10-31-93	Final 09-01-91 to 08-31-93

4. TITLE AND SUBTITLE

Spread Spectrum Random Access Networks

5. FUNDING NUMBERS

N00014-91-C-0234

6. AUTHOR(S)

Sastry, A. R. (Principal Investigator), Baker, J. E., Shahnawaz, I., and Clare, L. P.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Rockwell International Science Center
1049 Camino Dos Rios
Thousand Oaks, CA 91360

8. PERFORMING ORGANIZATION REPORT NUMBER

SC 71057.FR

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Sponsoring Agency: Ballistic Missile Defense Organization (formerly Strategic Def. Ini. org.)
Monitoring Agency: Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000
Contract Monitor: Dr. Rabinder Madan, ONR-1114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official BMDO/ONR position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A new method of code allocation in Code Division Multiple Access is proposed, based on partitioned information 'classes' (CDMA-IC), that is particularly efficient in meeting multicasting needs. Transmitters choose spreading codes based on the types of messages they are transmitting, and receivers listen to codes corresponding to the types of messages they are interested in. The performance of this approach can exceed that of both the traditional receiver-directed codes or broadcast codes in environments where there are several multicast groups (in which a given message is of interest to more than one node), or where it is difficult to know what types of messages (out of a fixed set of message types) a receiver is interested in at any particular time. Performance is further enhanced when receivers can simultaneously attempt to correlate multiple spreading codes. A broad theoretical framework was developed for the CDMA-IC concept and numerical results through simulation were obtained and compared to receiver-directed and broadcasting scenarios.

14. SUBJECT TERMS

Spread Spectrum, Random Access, code division, Multiple Access, Information classes, code allocations, Receiver-Directed assignments, broadcasting, Message Interest level

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Rockwell International
Science Center

Contents

List of Figures

Preface

This report is concerned with the development and evaluation of new ways of exploiting spread spectrum multiaccess capability for multicasting needs that would be useful in the ballistic missile defense environment. For this purpose, we developed a new scheme that we call Code Division Multiple Access with 'Information Classes' (CDMA-IC), in which, multiple access codes are assigned based on classes into which messages or information space is partitioned, rather than use the conventional receiver-directed method of code allocation. The new CDMA-IC scheme that we proposed will be highly beneficial in meeting the multicast needs of a set of nodes in which more than one node is interested in a given message and information needs to be disseminated rapidly and simultaneously in response to traffic surges. During 1991-93, under the Ballistic Missile Defense Office (BMDO, formerly known as Strategic Defense Initiative Organization, SDIO) Contract N-00014-91-C-0234, managed by the Office of Naval Research (ONR), we carried out detailed investigations in developing the CDMA-IC scheme concept. Simulations also were carried out to verify the concepts and to obtain quantitative results and compare them with that for receiver-directed and broadcast schemes.

The investigators would like to thank Dr. Rabinder N. Madan of the Office of Naval Research for his enthusiastic support of this research and interest in the subject matter. They would also like to thank a number of their colleagues at the Rockwell International Science Center and other researchers in the field, too large a list to be mentioned here, for their comments and helpful discussions.

Executive Summary

This report describes work on new and efficient spread spectrum random access network protocols for multicasting under the BMDO contract N00014-91-C-0234 managed by the Office of Naval Research, during September 1991 - August 1993.

The communications needs for tactical and strategic ballistic missile defense involve survivable interconnections of a large number of nodes in both space and on ground with highly transient traffic characteristics and processing loads. Such systems also often require multicast communications involving rapid dissemination of information among groups of nodes that are interested in similar messages and database updates. The combined multiple access properties of random access protocols together with spread spectrum techniques are very attractive for meeting such needs as random access is very efficient at light traffic loads while spread spectrum smoothens the impact of congestion when traffic increases rapidly and provides anti-jam protection in addition to low probability of interception. Though spread spectrum techniques are well understood in the literature for applications to anti-jam protection on individual links, efficient exploitation of its multiple access capabilities in a network environment have not been fully explored. Our investigations addressed development of new and efficient network protocols using spread spectrum concepts, with emphasis on applications involving multicasting and to situations in which information needs to be disseminated rapidly in response to traffic surges.

Under this contract, we developed a new protocol scheme that we call Code Division Multiple Access - Information Classes (CDMA-IC). In this scheme, all information is partitioned a priori into non-overlapping 'classes' and transmitters choose spreading codes based on the message classes they are transmitting, while receivers listen to codes corresponding to the classes of message they are interested in. All the receivers that are interested in the message of a given class receive it simultaneously, instead of requiring separate transmissions as in the case of the traditional receiver-directed codes. We have also examined through simulation appropriate power control methods to combat near-far effects due to mobility of nodes.

We have developed a theoretical framework to describe the concept in detail and various ways in which the schme could be tailored for different scenarios and could be parameterized appropriately. Also, through simulation results, we have demonstrated that the performance of the CDMA-IC approach can exceed that of both receiver-directed codes or broadcast codes in environments where there are several multicast groups (in which a given message is of interest to more than one node), or where it is difficult to know what types of messages (out of a fixed set of message types) a receiver is interested in at any particular time. Figure 1 shows a comparison of these schemes [1]. Performance can be further enhanced when receivers can simultaneously attempt to correlate multiple spreading codes. The concept of CDMA-IC scheme will be very useful in configuring BMD communication architectures in which multicasting is required.

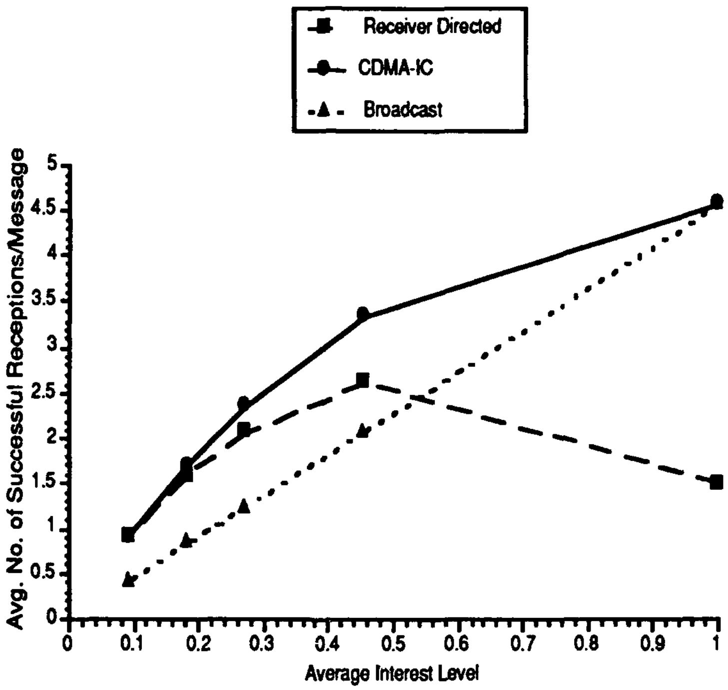


Fig. 1. Comparison of CDMA-IC with receiver-directed and broadcast modes.

During these two years, we developed a framework for efficient multicasting through CDMA-IC scheme. There are a number of directions in which the concept needs to be further expanded and analyzed, such as development of general theoretical models for CDMA-IC schemes considering multiple correlators, dynamic class partitions and code assignments for non-homogeneous traffic scenarios, and extensions to multi-hop spread spectrum networks.

I. Introduction

The communications needs for tactical and strategic ballistic missile defense involve survivable interconnections of a large number of nodes in both space and on ground with highly transient traffic characteristics and processing loads. Such systems also often require multicast communications involving rapid dissemination of information among groups of nodes that are interested in similar messages and database updates. The combined multiple access properties of random access protocols together with spread spectrum techniques are very attractive for meeting such needs as random access is very efficient at light traffic loads while spread spectrum smoothens the impact of congestion when traffic increases rapidly and provides anti-jam protection in addition to low probability of interception. Though spread spectrum techniques are well understood in the literature for applications to anti-jam protection on individual links, efficient exploitation of its multiple access capabilities in a network environment have not been fully explored. Our investigations addressed development of new and efficient network protocols using spread spectrum concepts, with emphasis on applications involving multicasting [1].

Spread-spectrum techniques [2]-[6] deliberately employ bandwidths that are much larger than the underlying information and are primarily used in military communications for security and for resistance to jamming and interference, though they are also finding increasing acceptance recently in commercial communications. Spread spectrum can also offer increased reliability in the presence of multipath and frequency-selective fading. Performance of direct-sequence (DS), frequency-hopping (FH), and time-hopping (TH) types of spread spectrum techniques have been extensively investigated in the literature considering a variety of jamming situations, modulation methods, and noise conditions.

Although spread spectrum is often used for anti-interference in a point-to-point link, it has high potential for use in multiaccess/broadcast type media so as to utilize its inherent capability to partition the total communication channel resource so that multiple transmissions (to multiple receivers) may take place simultaneously. In a typical receiver-directed system, each receiver is assigned a unique pseudo random (PN) sequence to realize the spread spectrum operation. The code set is to be chosen to meet low cross-correlation and high auto-correlation requirements [7],[8]. This capability of allowing simultaneous transmissions is referred to as Code Division Multiple Access (CDMA), and may be compared to other fixed assignment multiple access techniques, specifically Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). However, CDMA may not be as bandwidth efficient as FDMA or TDMA under some traffic scenarios if the sole objective is to partition the communications channel into some fixed number of continuous subchannels. (Since the code set typically only yields quasi-orthogonal subchannels, some error correction is usually also needed.) Nevertheless, the anti-interference characteristics of spread spectrum in combination with its CDMA capability make it very appealing for multiple access applications.

If each individual user generates traffic in a bursty (high peak-to- average ratio) fashion, then each user transmits information packets only when there is data to send. In packetized communication applications, synchronization of the PN code should be achieved at a receiver within a reasonable fraction of the packet length [2]-[6],[9] and with low false synchronization probability [10]. When communication among users in a multi-user channel is bursty, CDMA may be superior to fixed assignment TDMA or FDMA. The size of the user population may be far greater than the maximum number of simultaneous subchannels, and yet essentially all communications will be supported provided the number of active users at any instant is small enough. With such traffic requirements, it is proper to compare CDMA to demand-driven multiple access algorithms.

The communication needs of source-destination pairs often require a true *networking* capability, i.e., the ability for each user to communicate among many possible destinations at different times. Demand-driven multiple access algorithms must therefore determine who has a message to send and to which destination, and resolve any possible contention among simultaneous demands, in addition to permitting the actual data transfer to take place. Two basic types occur: Collision-avoidance protocols, such as token-passing, and random access protocols. Reservation schemes are also possible, in which the channel is partitioned into two parts, one for determining and resolving traffic demand, and the other for actual data transfer; the first part is achieved using preassignment, collision-avoidance or random access. It is the switching among source-destination pairs that requires us to consider CDMA in combination with another multiple access algorithm. We investigate spread spectrum in combination with random access schemes, since they are combined more naturally and are not as dependent on propagation delays as the collision-avoidance algorithms. Random access schen.es such as ALOHA-type schemes [11]-[15], carrier sense multiple access schemes (CSMA) [16],[17], and the splitting (or tree) algorithms [18]-[24] are natural for consideration in combination with spread spectrum since packets that collide in time can be retrieved to some extent.

The communications capacity is maximized by requiring that all message transmissions are of equal length, but variable length transmissions are possible at lower capacity. Requiring all messages to have the same length is unrealistic, as message length will be a function of the information contained, which is variable. One way around the fixed length constraint is to use ALOHA for a reservation subchannel, and then use a separate data channel on a reserved basis for the variable length messages. However, this approach is more susceptible to errors, particularly in a distributed control architecture, and more importantly, it requires an additional reservation delay. Thus it is likely that use of variable length messages on the random access channel is necessary to approach the optimal operation. The analysis of ALOHA schemes with variable length packets is quite complicated and often only bounds can be derived through approximate analyses [25]-[27].

The ALOHA random access protocol has a number of virtues, such as simplicity in implementation and a consequent robustness to errors and dynamics. However, use of ALOHA can result in unstable behavior, causing low throughput and excessive delays, unless an adequate control procedure is employed. There has been much work in the literature on analyzing stability control procedures for the slotted ALOHA system [28],[29], and development of control procedures to achieve stability for spread spectrum in combination with random access schemes [15], [30]-[36].

Another aspect worth reiterating is the importance of analyzing unslotted types of random access algorithms, since it s apparent that there is relatively little to be gained in slotting the system given that spread spectrum is being used [37]. Generally, little has been done in investigating control procedures and consequent stability measurements for unslotted systems. A related work is that of Molle [18], in which extension of tree algorithms to asynchronous (unslotted) operation is described. Dynamic control procedures for the ALOHA protocol must also be considered in an error-prone system. Retransmission control schemes have been investigated for the ALOHA protocol to yield stable performance when the channel is subject to errors [38].

The objective of this work is to study the performance of spread spectrum communications used in combination with random access algorithms in a network with multicast applications. Such techniques are very attractive for supporting communications among mobile stations with bursty (high peak-to- average ratio) traffic. As was stated previously, random access schemes are natural for consideration in combination with spread spectrum since packets that collide can be retrieved to some extent [11]-[15]. Use of spread spectrum in random access schemes also allows superposition of acknowledgment traffic on the same channel with only a marginal degradation in the overall throughput. In a bursty traffic setting, CDMA usually uses receiver-directed codes, so that each receiver listens only for its particular code and any user wishing to communicate to it must transmit on that code. In this sense, CDMA operates much like FDMA; note that if multiple transmissions are made to the same receiver then a collision occurs.

Our investigations addressed development of new and efficient network protocols using spread spectrum concepts, with emphasis on applications involving multicasting and to situations in which information needs to be disseminated rapidly in response to traffic surges. Under this contract, we developed a new protocol scheme that we call Code Division Multiple Access - Information Classes (CDMA-IC). In this scheme, all information is partitioned a priori into non-overlapping 'classes and transmitters choose spreading codes based on the message classes they are transmitting, while receivers listen to codes corresponding to the classes of message they are interested in. All the receivers that are interested in the message of a given class receive it simultaneously, instead of requiring separate transmissions as in the case of the traditional receiver-directed codes. We have also examined through simulation appropriate power control methods to combat near-far effects due to mobility of nodes.

In this report, we compare the performance of CDMA-IC to both receiver-directed (CDMA-RD) and broadcast approaches. In section 2, we will decsribe a theoretical framework of the CDMA-IC concept. In section 3, we present a description of the communication network model used in the analysis. Section 4 gives a description of the simulation approach used. In section 5, we present performance comparisons through numerical results for selected cases. Section 6 gives concluding remarks and identifies areas in which further work is needed.

2. CDMA-IC Scheme

The main emphasis in our work in this area is on the development of new, integrated performance measures that characterize both the anti-jam protection and the multicast capability of spread spectrum random access schemes. The conventional method of code allocation in Code Division Multiple Access (CDMA), receiver-directed code assignment, allocates a single quasi-orthogonal spreading sequence to each receiver. Transmitters choose the spreading sequence assigned to the selected receiver when transmitting a message. This approach works well when most messages are of interest to a single receiver, and transmitters know the identities of the receivers that would be interested in a particular message. It has the advantage that multiple transmitter-receiver pairs can communicate at the same time.

When the fraction of messages that are of interest to multiple stations increases, or when the identities of interested receivers are unknown, systems typically use a broadcast code, where all transmitters and receivers use the same spreading code. This works well when most receivers are interested in most messages, or when the traffic level is sufficiently low. However, when there are several disjoint multicast groups and traffic levels are fairly high, receivers can be blocked from getting messages they are interested in because they may be busy receiving messages that they will end up discarding.

In many environments, particularly military environments, messages belong to fixed classes, or types. The types of messages a node transmits or is interested in receiving can change over time, depending on the state of the node, or the state of the environment in which it operates. Alternatively, nodes may be members of one or more multicast groups that change over time. In this report, we suggest a new approach called "CDMA-Information Classes (CDMA-IC)," in which physical connectivity is matched to the desired logical connectivity in order to attain an efficient communication system. It provides a means to avoid the disadvantages of receiver-directed systems: the need for multiple transmissions to achieve logical multicast, and for the transmitter to know which receivers will be interested in each message, while at the same time avoiding the disadvantage of using a broadcast code and thus losing much of the simultaneous transmission capabilities of CDMA. Other advantages of CDMA-IC include (i) the ability to handle time-critical or bursty data of interest to multiple nodes;

(ii) the ability to support multiple baseband rates associated with each class, and (iii) smaller synchronization code sets than for receiver-directed CDMA in some situations.

Consider the transmission of a message between two individuals. This message relates to some particular topic among a set of many possible subject areas of conversation. Depending on the application domain, this information space may generally have a complicated structure with high dimensionality and cardinality. However, for our purposes we will assume a simple linear structure and finite extent, so as to ease the presentation of the concepts.

Specifically, we assume that the information space is divided into a finite number K of "information classes". Thus, each message is uniquely associated with one of the K information classes that identify the category or topic of information that the message deals with. The classes are denoted $C_1, \ldots, C_K$. We further simplify the space of information classes by assuming the classes may be envisioned as corresponding to evenly spaced points on a circle, and the distance between two classes is the shortest arc length between them. A possible application that may relate to this example is a defensive perimeter, perhaps centered on an aircraft carrier, in which each class is associated with a sector, i.e., information is tied to the physical locations of the objects. This is a very specific example structure, and different applications may have vastly different structures, however, this example is sufficient to introduce the concept. Another possible example is the set of preplanned types of messages used in a tactical context such as for logistics, maneuver control, and intelligence.

Note that while we consider the geometry of the information space, we have not yet introduced any notion of relative importance among the classes. The value of the transmission of a particular message certainly depends on who receives the message. For example, a message multicast to two recipients may have no affect on one and a large affect on the other. This is due to the different interests and responsibilities among the set of nodes (users) in the system. Therefore, for each node, we will associate a number for each information class that quantifies the "level of interest" that that node has as a recipient of a message that belongs to that class. This level of interest not only captures the relative affect on the consequent behavior of the node, but also the relative impact of that behavior on the overall objectives of the mission.

It is also useful to identify a level of interest for each node as a potential *transmitter* of a message of each class. This may indicate the relative impact that that node is likely to have. A closely related notion is that this quantifies the quality or reliability of the message content. Another possible interpretation for this parameter is that it is inversely related to the cost for that node to transmit the message.

The actual value achieved by the delivery of a particular message is measured as a function of the two parameters: one associated with the level of interest of the transmitter, the other associated with the level of interest of the receiver. We now formalize these notions.

Denote the number of nodes in the system (network) by N. The nodes are denoted by i, i=1,...,N. Associated with each node i and each class C_k, $k = 1,...K$, is the parameter s_{ik} that denotes the relative value of node i transmitting a message in class C_k, as well as the parameter d_{ik} that denotes the relative value of node i sucessfully receiving a message in class C_k. We assume $s_{ik} \geq 0$ and $d_{ik} \geq 0$ for all i and k.

The value of node i transmitting a message in class C_k to node j is determined by a nonnegative value function $v = v(.,.) = v(s_{ik}, d_{jk})$. A possible choice for the value function v is $v(s,d) = sd$. Many other choices may be selected as appropriate for application domain targeted by the analytical model.

We now utilize the aspect of our model that the information classes form a metric space. We assume that for each node i, the *distribution* of that user's level of interest over the set of information classes is linked to the structure of the underlying metric space. Note that the users may generally all have different interest level distributions, but are nevertheless related to some degree by this common structure. It is useful to extend the linear example to convey the concept. Suppose that for a particular node, the density of its level of interest as a receiver is portrayed by Figure 1, defined as d_{ik} where i is fixed and the abscissa varies over the set of classes $\{k=1,...,K\}$. This function has a single parameter a, such that as a tends to unity, the density is uniform over the entire universe of information classes, while as a tends toward zero the density concentrates as a point mass on the class at the center point, as shown in Figure 2.

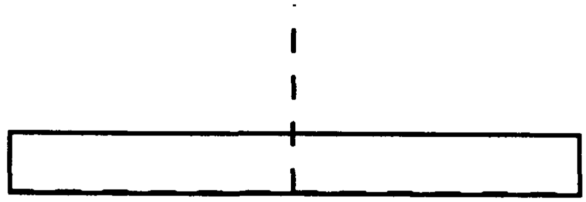


Figure 2a. Density with $a = 1$.

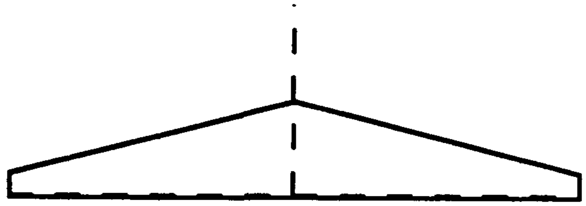


Figure 2b. Density with $a = 3/4$.

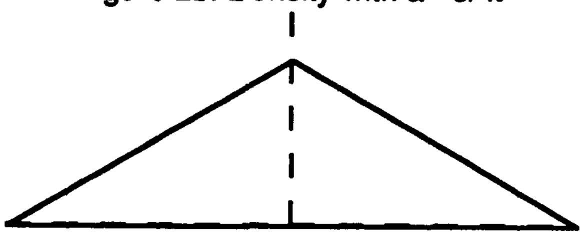


Figure 2c. Density with $a = 1/2$.

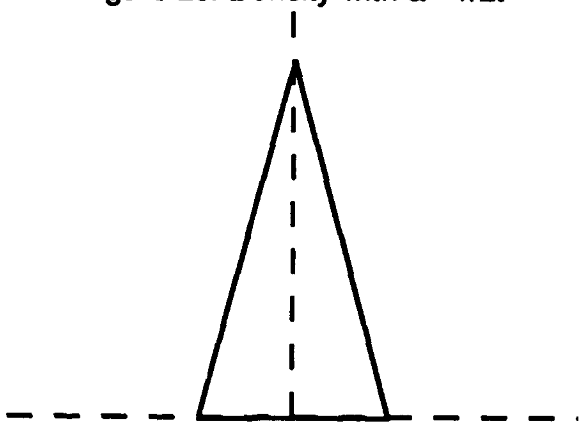


Figure 2d. Density with $a = 1/4$.

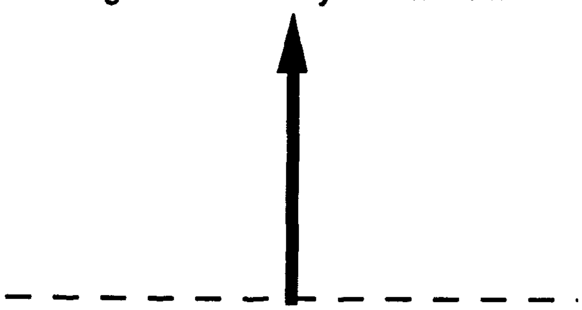


Figure 2e. Density with $a = 0$.

Figure 2a-2e. Probability density functions of level of interest in a message

The function of Figure 2 may be extended to allow translation of the density by another parameter denoted as *b*, making use of the cyclic assumptions on the distance function. Thus, *b* is the class that the node is most interested in. Example cases illustrating such a translation are provided in Figure 3.

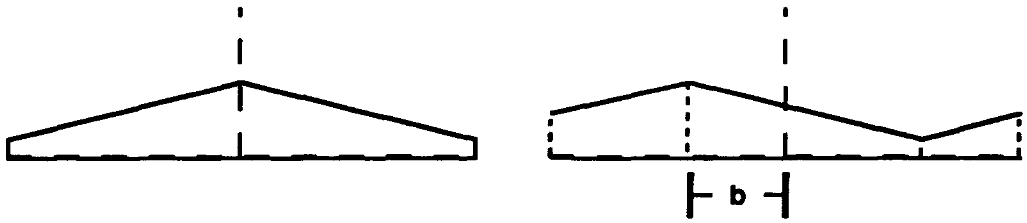


Figure 3a. Translation of the case with *a* =3/4.

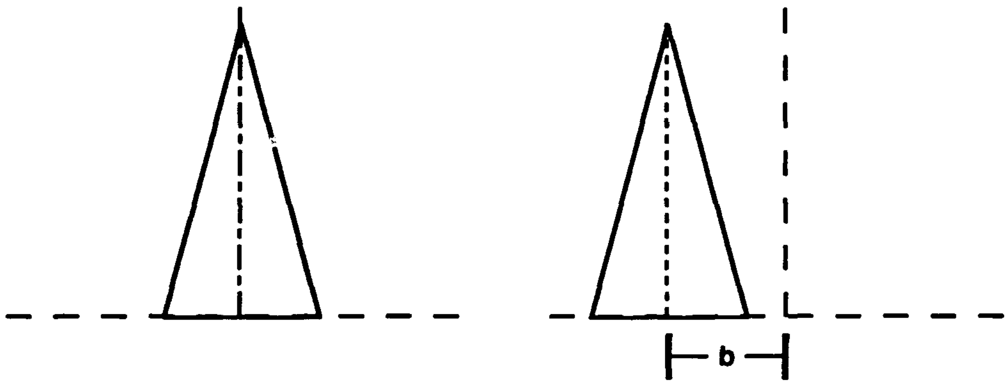


Figure 3b. Translation of the case with *a* =1/4.

We further extend the family of level of interest density functions using one more parameter *c* that simply scales the overall density. Each density is assumed to be normalized such that $d_{ik} = c\lambda_k$ for all *i* and such that $\sum k\lambda_k=1$ (and $\lambda_k \geq 0$). Thus, one can view $\{\lambda_k\}$ as a probability density having the two parameters *a* and *b*. The parameter *c* models the relative overall importance of the node. The receiver level of interest for a particular node *i* is determined by choosing specific values for the parameters *a*, *b*, and *c*.

The same three-parameter family of density functions indicated above (and by Figures 2 and 3) may be used to model the nodes' levels of interest (reliability etc.) as transmitters: $\{k\text{: } s_{ik}\}$ for each *i*. To distinguish the parameters for the transmitter level of interest (LOI) from those of the receiver LOI we denote the chosen transmitter variables as *a'*, *b'*, and *c'*.

Suppose that node i has a potential message to transmit that belongs to class k. If node i's transmitter level of interest in class k is zero (s_{ik}=0) then since the value in its delivery $v(s_{ik},d) = s_{ik}d = 0$ for any d, there is no point in node i's transmitting the message. For example, if node i's transmitter level of interest density has the form of Figure 2c, then there are many such classes. Therefore, in the definition of the message generation process, we might as well assume that node i does not generate any messages for which s_{ik}=0.

There is no a priori restriction on the message generation processes. However, it appears reasonable that the rate at which node i generates new messages belonging to class k would be related to its transmitter level of interest s_{ik}; the rationale is essentially an extension of the special case of s_{ik}=0 discussed above. Based on this logic, we suggest that the traffic generation processes may be simply defined as a set of NK independent Poisson point processes having production rates rs_{ik}, $i=1,\ldots,N$, $k=1,\ldots,K$. The parameter r allows the user of the model to vary the overall traffic load offered to the system, given as $r\sum_i c_i'$.

Summarizing, the system model is defined by the information classes denoted $\{C_1,\ldots,C_K\}$, the nodes denoted $\{i: i=1,\ldots,N\}$, a set of N 3-vectors $\{(a_i,b_i,c_i)\}$ that define the nodes' receiver levels of interest, a set of N 3 -vectors $\{(a'_i,b'_i,c'_i)\}$ that define the nodes' transmitter levels of interest, and a parameter r that governs the overall rate of message generation. Performance is determined by the accumulation of values of all received messages, where a message belonging to class C_k that is received by node j that was transmitted by node i will contribute a value of $v(s_{ik},d_{jk})=s_{ik}\,d_{jk}$. Note that multicast messages are allowed, so that the value of a particular transmission is summed over all nodes that successfully received the message in that transmission. Of course, reception of a duplicate message is worthless, should the network protocol (not yet discussed) cause duplicate transmissions to occur for a receiver.

The number of free parameters may be further reduced while still allowing sufficient generality for investigations of the CDMA-IC concept. We may restrict $a_i = a$ and $a'_i = a'$ for all i, for constants a and a', so that the variances of each node interest over the classes about its mean is the same. We may also take $c_i = c'_i = 1$ for all i, so that all nodes are equally important/reliable and generate the same average amount of traffic. The remaining level of interest parameters $\{b_i,b'_i\}$ determine how the nodes place emphasis on different areas of interest. They may be symmetrically distributed by setting $b_i = b'_i = 1$ for $i = 1,\ldots,N/K$, $b_i = b'_i = 2$ for $i = N/K+1,\ldots,2N/K,\ldots$, and $b_i = b'_i = K$ for $i = (K-1)/N+1,\ldots,N$.

We now may make several observations regarding the selection of codes for the underlying CDMA system. Suppose for now that $a' \equiv 1$ (see Figure 2a), so that all nodes' transmitter level of interest and message generation rates are uniform over all the classes. If in addition $a \equiv 1$, then every message is equally valuable to all the nodes, so that clearly a single broadcast code would be optimal. At the other extreme, if $a \equiv 0$ (see Figure 2e), then K quasi-orthogonal CDMA codes should be employed, one for each class ("class-directed" codes; in the case $N=K$, this becomes "receiver-directed"). Intuitively, the intermediate case where $a \equiv 1/2$ (Figure 2c) is optimized for some number of codes greater than one but less than K. Precise quantification of this result is an open problem and will depend on specific assumptions on the channel model and access protocol used. However, the extension of our simulation to incorporate the above model would be quite straightforward, along with the characterization of results such as these.

3. Communication System Model

In this section, we describe the communication system model that we used to evaluate and demonstrate the CDMA-IC concepts developed above. These system features have been included in a simulation model that is described in detail in the next section. Though the CDMA-IC framework described in section 2 has a variety of features that require detailed investigations, we evaluated only a simplified version with a limited set of features through simulation. We model a collection of nodes communicating through a physical broadcast radio channel. Nodes use a pure, unslotted random access protocol to access the channel. Packets are transmitted using BPSK, where the spreading code is selected according to the transmission scheme (see below). The radio channel model includes propagation power loss, noise, and multiuser interference. Each communication node has five major subsystems: (i) a node state model, (ii) a message traffic subsystem, (iii) a movement subsystem, (iv) a timing subsystem, and (v) a communication subsystem, as shown in Figure 4.

3.1 Communications Node State Model

Each node has a state, which determines the types of messages the node generates and the rates with which they are generated, and the types of messages that the node is interested in receiving. These states could correspond to a node searching for targets, forwarding targeting information, negotiating firing assignments, and so forth. The node state can change either at random times or in response to an incoming message.

Each node starts in an initial state, and then (ignoring, for the moment, the effects of incoming messages) moves from state to state according to a continuous time Markov chain. When a node enters a new state, it will remain in that state until either an exponentially distributed delay elapses, or until a message causes a state transition. The mean of the exponential delay depends upon the node's state. If the delay elapses, the node selects a new state according to probabilities determined by its current state.

The node state can also change due to an incoming message. Whenever a message is successfully received by a node (i.e., it is a message in a class of interest to the node, given its current state, and was correctly received) the node can, with a

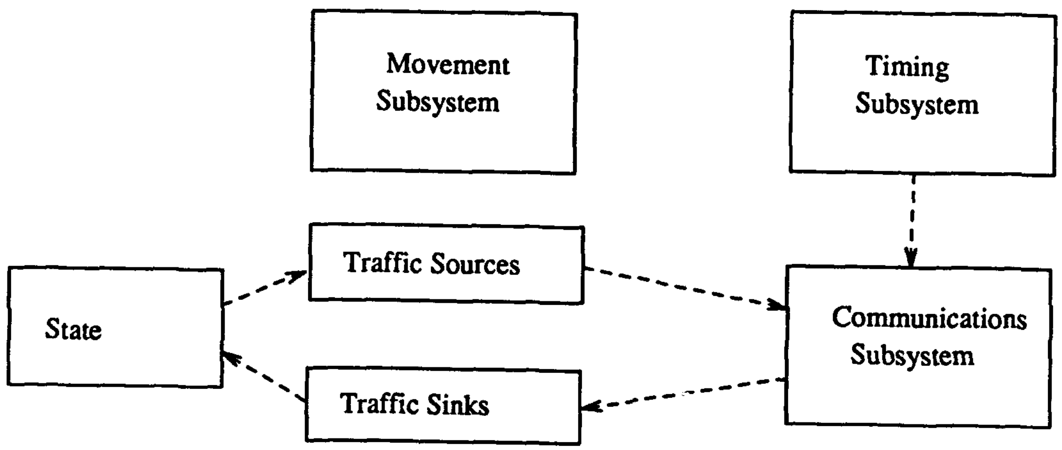


Figure 4. Components of node

probability determined by its current state and the incoming message class, move to a new state.

3.2 Message Traffic Subsystem

The node generates messages according to a random process. The message generation rate and the classes of the generated messages depend upon the node's current state. In particular, message intergeneration times are exponential random variables with a mean that depends upon the current state. When a message is generated, its class is chosen randomly, and depends upon the node state (i.e., for each state there is a vector of probabilities indexed by message class).

The number of classes of messages that a node is interested in receiving, and their identities, depend upon the node's state. Messages do not have specific nodes as destinations: they are intended for all nodes that are in a state in which they are interested in messages of that class.

In each state, a node is interested in receiving messages of certain classes. This is determined by an input binary array which is indexed by node state and by message class. If the node is in state i and it is interested in messages of class j, then the (i,j) th entry will be a 1, otherwise it will be a zero. We allow nodes to be interested in multiple classes of messages at the same time, and allow nodes to be interested in a particular class in several states.

3.3 Movement Subsystem

The movements of each node through 3-dimensional space are specified by a series of waypoints and speeds. A node moves from waypoint to waypoint, following the great circle containing the projections of the two points, at constant speed. Node positions and relative velocities are used to calculate propagation power loss, propagation delay, and Doppler shift. Each waypoint consists of latitude, longitude, and altitude. We assume a spherical earth when converting the waypoints to Cartesian coordinates.
A user can specify movement profiles to individual nodes whether random motion or projected trajectories.

3.4 Timing Subsystem

Each node has a separate clock. We model clock drifts and clock updates using an external source such as Global Positioning Satellites. Clock drifts are modeled as random, zero mean shifts which occur at random times. The statistics are inputs to the simulation.

3.5 Communication Subsystem

The communication subsystem consists of the channel model, the transmitter and the receiver, as shown in Figure 5. The communication subsystem alternates between transmitting, attempting to acquire incoming signals, synchronizing to an incoming transmission, and receiving a transmission as shown in Figure 6. Interfering transmissions (whether inadvertant or deliberate) and noise are used to determine the probability of achieving synchronization with an incoming transmission, and to determine the bit error rate seen during the transmission. The BER can change during the course of a transmission if, for example, an interfering transmission reaches the receiver. At each such point, the BER is recalculated and used to determine the distribution of the number of errors during the next transmission segment.

3.5.1 Channel Model

We model the channel effects of power loss due to propagation, receiver loss, and antenna gains; propagation delay; and doppler shift. We apply the channel effects at the receiver, taking into account the position and velocity of the transmitter (known to the simulation). For free space propagation, the received power, P_R, is given by [4]:

$$P_R = \frac{P_T G_{TR} G_{RT}}{(4\pi R / \lambda)^2 L_C}$$

where,
P_T = transmitted power;
G_{TR} = gain of the transmitter's antenna in the direction of the receiver;
G_{RT} = gain of the receiver's antenna in the direction of the transmitter;
R = range between the transmitter and the receiver;
λ = center wavelength of the transmission,
L_C = receiver loss factor.

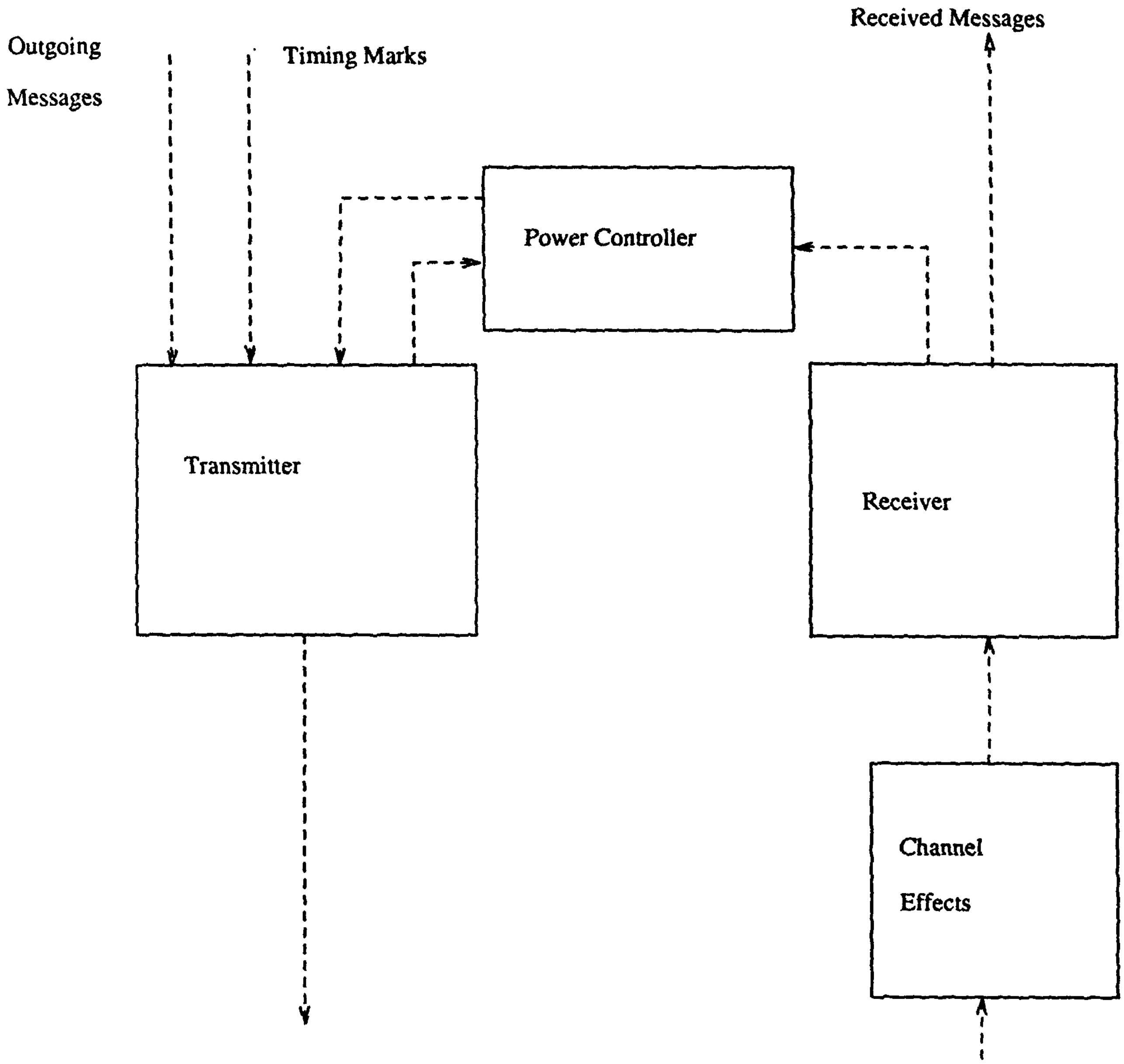


Figure 5. Communications subsystem

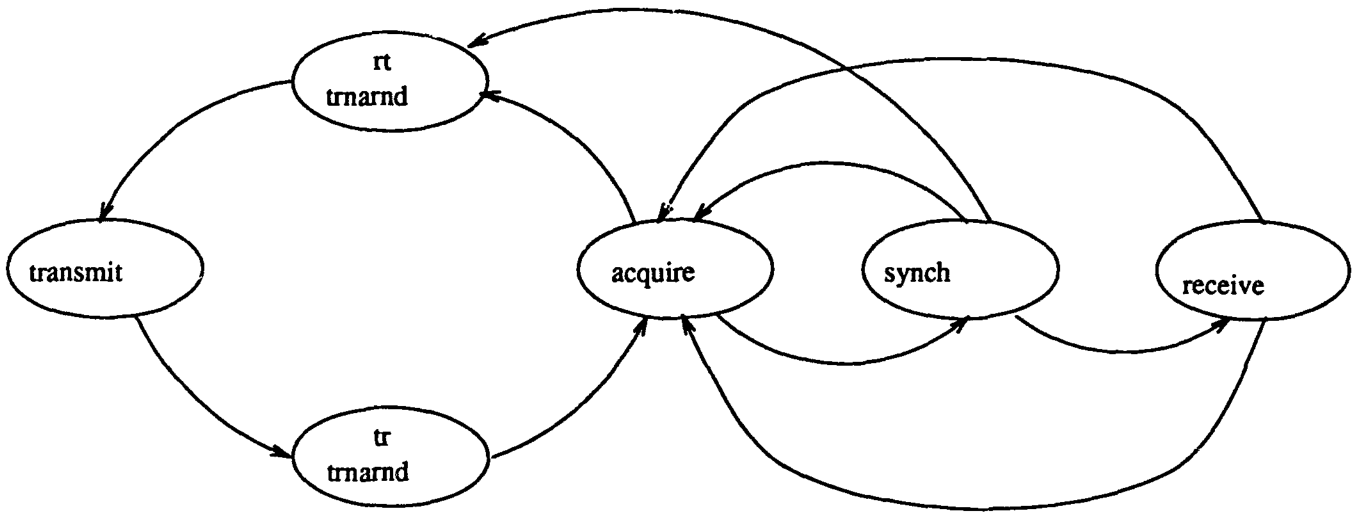


Figure 6. Communication subsystem states

3.5.2 Tarnsmitter

We consider three different transmission schemes: receiver directed codes, broadcast codes, and information class codes.

Receiver Directed Codes: In the first scheme, transmission and reception are by receiver directed codes. When a message is generated for transmission, the transmitting node looks at the message class, and estimates which set of nodes is currently interested in receiving messages of that class. We make the optimistic assumption that the transmitting node knows the current states of all possible receiving nodes exactly. The transmitting node makes a separate copy of the message for each interested node, and sequentially transmits them using receiver directed codes.

Broadcast Code: All transmissions use the same broadcast code. Nodes discard messages they receive that they are not interested in. However, we assume that nodes are not able to determine whether they are interested in a message until after they have completely received it.

Information Class Codes: The message spreading code will be based solely on the message's class, and no specific receiver will be chosen. It will be received by any node that is trying to synchronize to that message spreading code.

Transmitters use an unslotted ALOHA random access scheme. When a message is generated for transmission, if the node is not currently transmitting a message, or receiving or synchronizing to an incoming message, it will immediately begin transmitting the new message. Otherwise, it queues the message until it is able to transmit it.

3.5.3 Receiver

We assume that the receiver can attempt to acquire M spreading codes at the same time but can only receive one message at a time. This would be consistent with a RAKE-like receiver design. We allow for the case $M=1$. During the acquire state, any incoming message with any of the spreading codes that the receiver is attempting to synchronize to (in the case of information-based code, this means from any of the classes of interest) will initiate a synchronization attempt, and any message of any of those classes that arrives during a fixed length collision window will cause a collision. We currently use a simple threshold model for determining whether a signal is acquired. If a signal intended for a receiver arrives while the receiver is in 'acquire' state, it will be successfully acquired if its Doppler shift is within specified limits and its signal to noise ratio (including jamming and multiuser interference) is high enough. The receiver model considers thermal noise, jamming, multiuser interference, and received power level. It is based on Sousa's model [39]. The impact of multiuser interference at different power levels is modeled.

4. Simulation

We have simulated the network model using the Block Oriented Network Simulator (BONeS)* [40] simulation package. We use a combination of analytical and simulation techniques. We model node movement, message generation, and state changes as simulation events. We use analytical techniques to determine channel effects, probability of detection, and bit error rates between "events" which correspond to significant changes at a receiver (such as the start or end of an interfering reception, the change of power or bandwidth of a jammer, or slow channel fading). We assume that reception conditions are stationary between events.

The impact of multiuser interference at different power levels is calculated using Sousa's model [39] to determine the bit error rate during each part of the transmission which has a constant set of interferers and jamming level. We use a binomial distribution to sample the number of bit errors during each interval, and, summing, determine a total number of bit errors for a packet. If the number of bit errors is less than the number of correctable errors for the code in use, the packet is considered to have been received successfully.

A brief description of the simulation environment and the details of modeling the individual subsystems is given below.

4.1 BoneS Development Environment

The model is constructed via the usage of the BONeS (Block Oriented Network Simulator) Designer package [40]. This package, which consists of various 'pre-fabricated' software utilities and 'primitives' encapsulated in the form of 'Systems Approach' or 'Flow Chart' oriented blocks and logical constructs, enables simulation model designers to express their designs in a Modular Block Diagram form with elaborate 'Top Down' or 'Bottom Up' Schematic definition (to which their technical world views can intuitively relate). This contrasts from the traditional programming language statements which albeit can also be modularized and encapsulatled, but may still require several levels of translation in order to comprehend the significance of what they functionally represent.

---

* BONeS™ is a Trademark of Comdisco Corporation.

It is the BONeS Compiler that internally translates the properly constructed user produced block diagrams into C code and thus substantially relieves the designer from the major preoccupation with the various programming language implementation and syntax usage issues and memory management concerns, allowing greater focus on the technical application issues. However, it should also be pointed out that the BONeS package does have its own underlying 'syntax' and 'grammar' usage rules governing proper block diagram construction, elaboration, co-ordination, linkage with 'sub', 'peer' or 'parent' components, etc..Programming language practioners may view the package as a 'pictorialized' procedural language with a pseudo Object Oriented flavor.

The package also provides designers with the ability to innovatively apply the BONeS grammar to construct their own application specific primitive blocks and incorporate them into the package, so that simulation modeling capabilities can be expanded beyond what were initially provided.

Finally, the package also provides various output data gathering, reduction and visualization tools which enhance its power in aiding designers to rapidly verify simulation models and graphically represent output results.

4.2. Simulation of the CDMA-IC model

The model is capable of simulating the operation of a Mobile Spread Spectrum network of `nodes' that transmit `packets' over a simulated `medium' capable of carrying packets belonging to one or more `Information Classes'.

Additionally and independent of the number of Information Classes specified, each node has been provided to operate in two (user specifiable) modes: Single Correlator, or Multiple, i.e., `nc' Correlators, where `nc' represents the number of Information Classes of Interest to the nodes, as specified by the user.

Since this model is a superposition of three sub-models where each sub-model highlights one of the above aspects, this section is divided into three seperate subsections: (1) the first section describes the design of the information-class aspects of the model, (2) the second section describes the spread spectrum Communication aspects of the model assuming the presence of a single correlator in the transceiver of each node, and (3) the third section describes the extension to multiple correlators aspect of the Model

4.2.1 The Information Class Sub-Model

Each node is capable of assuming and transitioning through a set of user specified states, starting from an Initial State. The node is capable of remaining in each state for an `in-state duration' that is randomly selectable via a (user specified) statistical probability distribution, and further, the model provides the user with the capability to assign different `mean values' for the in-state durations for each different state via a `Mean In-State Duration Vector'. The Mean In-State Duration Vector is a real vector of length equal to the total number of defined states, and its elements are hence indexed by integer values representing state identification numbers. Each element then, represents the mean duration for which the node shall remain while in a particular state (provided it is not prematurely forced to transition to another state in response to any incoming message).

While in a particular state, the node is capable of generating transmission packets whose (random) inter-generation times (IGTs) are also associated with this particular state. Further, the model provides the user with the capability to generate different `in-state inter-arrival times' for different states via a `Message State IGT Vector' in a manner analogous to the above mentioned `in-state durations'. The `Message State IGT Vector' is a real vector of length equal to the total number of defined states, and its elements are hence indexed by integer values representing state identification numbers. Each element then, represents the mean message inter-arrival time for which the node shall generate messages while in a particular state.

Thus, there exists a one-to-one correspondence between the set of states and the set of Information Classes.

While in a particular state, the node transmits packets of a particular Information Class and at a rate corresponding to that state. This is achieved via a `Packet Transmission Rate Vector', which is a real vector of length equal to the total number of defined message information classes/node states, and its elements are hence indexed by integer values representing state/class identification numbers. Each element then, represents the packet Transmission Rate (actually the baseband Information Rate) in bits per second, at a node while in a particular state.

Additionally, the node may receive packets but only those of the Information Classes of Interest to this node. This is achieved via the assignment of an 'Acceptable State Message Class Binary Matrix' which is a rectangular matrix of size 'MxN', where 'M' is the number of states that a node can assume, and 'N' is the number of Message Classes that can be receiveded during a state. Further, each state is represented by an integer between '0' and '*M*-1', and each Message Class is represented by an integer between '0' and '*N*-1'. Thus, each element, '*B(i,j)*', of the matrix represents the binary value '*B*' (equal to '0' or '1') that a node currently in state '*i*' considers a message of Class '*j*' to be, acceptable ('1') or unacceptable ('0') for reception.

Upon completion of the an 'in-state duration', the node has the capability of randomly 'deciding' whether it will remain in the current state or transition to another state. The decision as to whether or not the node transitions is made via a user provided cumulative probability distribution associated with this current state. Further, the model provides the user with the capability to assign different (cumulative) probability distributions to each state via a 'Next State Probability Matrix', which is a square matrix of size '*MxM*', where '*M*' is the number of states that a node can assume, and each state is represented by an integer between '0' and '*M*-1'. Thus, each element, '*P(i,j)*', of the matrix represents the probability '*P*' that a node currently in state '*i*' transitions to a next state '*j*' (provided that the node was not forced to prematurely terminate its duration in state '*i*' in response to an incoming message).

Thus far, if its state transition operations expire 'naturally', i.e., without interruption by a critical message, the node transitions from state to state along a prescribed 'path of states', reamining in each for a pseudo- random duration determined by and transmitting and receiving/processing packets relevant to the state.

It should be noted that while in a particular state, all (relevant) received packets have the ability to cause the node to 'abruptly' change its present state and immediately be 'directed to transition to another prescribed state'.

When the node receives a packet and upon processing determines that the packet is 'relevent to its present state', the node decides 'how' it will react to this packet by generating a random number and comparing this number to the appropriate `reaction probability' assigned to the present state. The reaction probability is accessed from a user provided `Node Message Reaction State Probability Matrix' which is a rectangular matrix of size *`MxN'*, where *`M'* is the number of states that a node can assume, and `N' is the number of message information classes it can receive, and further, each state is represented by an integer between `0' and *`M*-1', while each message class is represented by an integer between `0' and *`N*-1'. Thus, each element, *`P(i,j)'*, of the matrix represents the probability *`P'* that a node currently in state `i' will `react', i.e.,transition to another state, in response to having received a message of class *`j'*. Due to its reactive nature, the transition will be immediate, forcing the node to prematurely terminate its current `in-state duration'.

Having determined that a `reaction' to a message is necessary, the node then consults a `Node Message Redirected State Matrix' to obtain the next state to which it will transition. The `Node Message Redirected State Matrix' is a rectangular matrix of size *`MxN'*, where *`M'* is the number of states that a node can assume, and *`N'* is the number of classes of messages it can receive, and further, each state is represented by an integer between `0' and *`M*-1', while each message class is represented by an integer between `0' and *`N*-1'. Thus, each element, *`S(i,j)'*, of the matrix represents the Next State *`S'* that a node currently in state `i' must transition to, in response to having received a message of class `j', when the message forces the node to `react' (i.e., transition to another state) and prematurely terminate its current `in-state duration'.

4.2.2 The Communications Sub-Model:

This section first outlines the operation of the network in the Single Correlator Mode, where a in this mode, the operation of a communications node (specifically, its transceiver) is characterized via the `communication state transitions' of a single `Communication State Machine' which transitions through the following set of communication states: Transmit Packet, Acquire Packet, Synchronize Packet, and Receive Packet. In addition, there are several sub-modules that perform checks to determine whether or not the node performed its processing functions successfully during the above states, particularly, the `Acquire' through `Receive Packet' States. The communication sub-model description, with the aid of references to appropriate block diagrams, now follows.

The detailed design of the model is shown in the Appendix in the form of BONeS block diagrams, which, in spite of being presented in a Top Down `Schematic' order, may still confuse the reader unfamiliar with the BONeS Designer package. Hence, an overall system block diagram is presented in Figure 7 with a brief explanatory overview given below that walks the readers through this diagram enabling the understanding of the model's operational design. The detailed schematics mentioned above are expansions of this diagram indicating various operations.

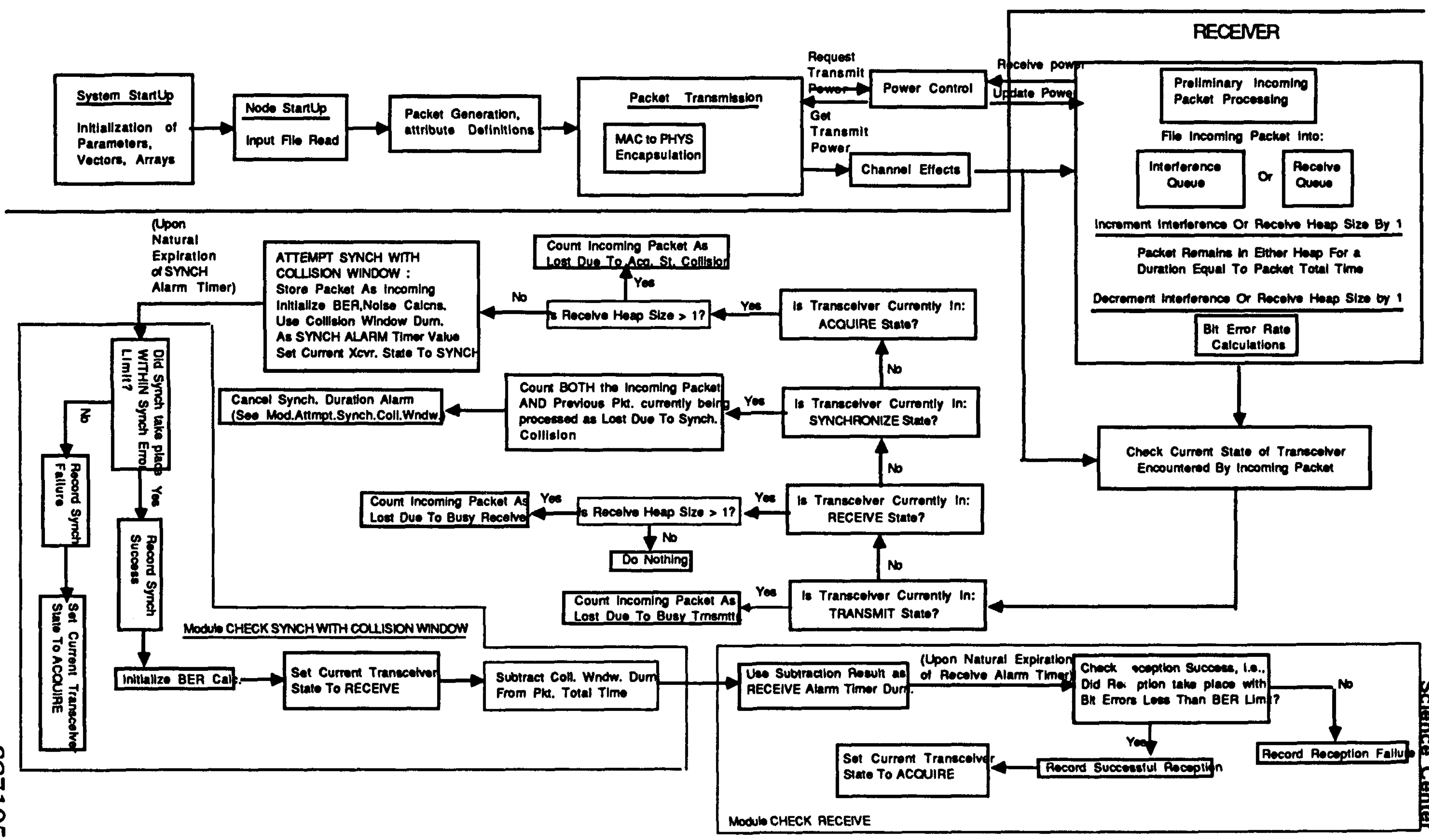
The Initialization Block represents the system and node startups, which involve user specified input data file access, parameter initializations and memory allocation functions performed at the very beginning of the simulation.

Following this, control is passed to the Nodes, whose Packet Generator submodules (with user specifiable inter-generation times, packet lengths and other relevant delays) are activated and each proceeds to create the data structures representing a `Packet' which is then defined, i.e., its attributes are assigned from the user specified parameters. The data structure of a typical packet is included in the Appendix.

The packet is then passed to the Transmitter submodule where it goes through further processing, such as in the Open System Interconnection model's layer encapsulations, along with further attribute definitions related to transmission mode, destination, and insertion of other user specified transmission related parameters including transmission power control. Depending upon the transmission protocol specified by the user, the packet is then `transmitted' i.e., passed to the submodule characterizing the `Medium' or `Channel Effects'. The block diagram of the Transmitter Submodule along with its associated submodules are included in the Appendix.

The Channel Effects submodule characterizes the propagation of the packet over the channel taking into account various effects due to the Node's Relative Velocity (if the Node is mobile), propagation power loss in free space or otherwise, attenuation, near-far effects, etc.. The packet is then passed to the Receiver submodules of all Nodes connected to the medium thereby characterizing broadcast effects over the channel. The block diagram of the Channel Effects Submodule along with its associated submodules are included in the Appendix.

Figure 7. System block diagram for the spread spectrum network simulation



The Receiver submodule characterizes the message filteration and reception processing of the packet and includes effects due to interference, noise, received power update, and also jamming and fading if specified by the user.

Due to the Receiver module behaviour modeld as a multi-state machine transitioning between packet Acquisition, Synchronization and Reception states, the packet processing at the Receiver is contingent upon the current communication state of the Receiver, and in conjunction with the operation of the Transmitter submodule, is capable of modeling half or full duplex transmission. Packet collisions (incurred when more than one packet arrive within the same `collision window' of a node's receiver) and other losses (incurred when an incoming packet encounters its destination receiver in an `inappropriate' communication state) are modeled. Determination of a `successfully' synchronized and/or received packet via computation of Bit errors are also modeled.

Appropriate determinations are made regarding whether or not an arrived packet's computed bit errors (during synchronization and then seperately during reception events) exceeded certain user specified error limits. The corresponding appropriate statistics associated with successful or failed events are also recorded and maintained for the duration of the simulation run.

At the end of the simulation run, an output results file is produced. The block diagram of the Receiver Submodule along with its associated submodules are included in the Appendix.

It should be pointed out that after initialization and depending upon the input data, each node operates independently of and asynchronously with respect to its peers, performing operations as directed by its own communication state machine.

4.2.3 The Extended Multiple-Correlators Sub-Model

This section extends upon the previous outline of the operation of the network in the Single Correlator Mode to the Multiple Correlator Mode.

If in the Single Correlator Mode, the operation of a node is characterized via the `communication state transitions' of a SINGLE `Communication State Machine', then the Multiple Correlator Mode operation is characterized via the same communication state transitions, but in an ARRAY of `n' Communication State machines PER NODE, where each communication state machine operates independently of and asynchronously with respect to its `siblings' in the SAME Node.

The foregoing description assumed that all nodes transmitted and received packets of the SAME (single) `Information Class' and that each node possessed a single Correlator.

When the model is set for execution with the option for the nodes to operate with Multiple Correlators, where the number of correlators per node depends upon the number of Information Classes specified by the user, then the network sets itself such that all its nodes utilize those components (such as transmitter and receiver) that are designed to handle multiple, quasi-parallel, transmissions and receptions. The multiple-correlator versions of the transmitter and receiver are each endowed with `n' (rather than one) different `Communication State-Machines' (where n is the number of Information classes).

The state transitions of each one of these individual, asynchronous and quasi-parallel state-machines then represent the operation of a corresponding correlator that is assigned to a specific information class. It should be noted however, that while in this mode each node can handle `n' seperate packets at a time (each of a distinct information class), packet collisions between two or more packets arriving within the `collision window' associated with the same information class are still possible and can be `detected'.

5. Numerical Examples

In this section, we will present numerical comparisons of the performance of CDMA-IC, CDMA-RD (receiver-directed), and broadcast CDMA. We consider a case with 12 communicating nodes statically assigned to multicast groups. Each node belongs to a single multicast group. Nodes do not change groups, and in the case of CDMA-RD, we assume that the transmitter knows the identities and spreading codes of all receivers in its group. This provides the best case performance for CDMA-RD. We used a simplified model compared to the broad features described in the CDMA-IC frmawork described in section 2, for the purpose of simulation, using the following:

The number of classes Classes, $\{C_i\}$, $i = 1, K$;
Number of nodes, N
States, $\{S_i\}$, $i = 1, L$ Represent allowed classes of Tx/Rx for Nodes
State transistions: Message Directed and Time-Directed
No. of classes of interest to node $n = I_n$, $n = 1, N$
Average Interest Level = $[((\sum_{n=1}^{N} I_n)/K)-1]/(N-1)$
Reception Scenarios

- Single Correlator - Single Class of interest
- Single Correlator - Multiple Classes of interest (1 message only rcvd at a time)
- Multiple Correlators - Not Simultaneously
- Multiple Correlators - Simultaneously

Results: various parameters Vs average interest level

We vary the number of groups from 1 to 6 and compare the performance of the three schemes. In Figure 8 we consider a high traffic case, while in Figure 9 we consider a low traffic case. We plot the results versus average interest level, which we define to be the average number of nodes interested in receiving a message, divided by the total number of possible receiving nodes. Thus, when all nodes are in a single group, the average interest level is 1; when each group consists of two nodes, the average interest level is 1/11. We assume that in the CDMA-IC case, separate information classes have been assigned to each multicast group. In the Broadcast case, all messages are transmitted with the same spreading code. Nodes are unable to determine whether they are interested in a particular message until after they have received and decoded it. Thus, nodes can be blocked from receiving messages they are interested in, because they are in the midst of receiving a message they are not interested in. In the CDMA-RD case, the transmitting node sends a separate copy of its message to each node in its multicast group, using the receiving node's spreading code. We consider a case with a baseband transmission rate of 16Kbs and a direct sequence spreading bandwidth of 10MHz.

In Figure 8a, we see the variation in the number of successful receptions per message as a function of average interest level. We see that Receiver-Directed operation achieves a higher number of successful receptions per message for low levels of average interest, while Broadcast does better for high average interest levels. Note from Figure 8b that the average delay increases for Receiver-Directed when the average interest level increases, due to the need for the transmitter to send the message separately to each receiver. We see that CDMA-IC outperforms both CDMA-RD and Broadcast across the entire range of interest levels, while matching the low delay of Broadcast.

In Figures 8c-8f, we consider a single reception, and break down the probabilities of successful reception and of loss into their constituent parts. We see that in the broadcast case, most losses are due to the receiver being busy, or packet collision, while in the CDMA-RD case, losses are due to the receiver being busy.

In Figure 8g, we look at the average number of successful receptions per transmission, and see that CDMA-IC again outperforms both CDMA-RD and Broadcast, and provides a smooth transition between the two as the average interest level changes. Figures 9a-9g give corresponding curves for the case when the traffic level is made lower by increasing the mean packet intergeneration time to 1 sec. Though quantitatively different, the curves in Figures 9a-9g qualitatively exhibit similar characteristics as in the case of Figures 8a-8g explained above, demonstrating the superior performance of CDMA-IC scheme when a set of multicast groups are involved.

Figures 10 and 11 show the impact of using multiple correlators (two in our example). When a node is interested in two classes but has only a sigle correlator available, it will lose half the packets of interest, as shown. However, when two correlators are available, the performance is influenced further by whether the hardware struture of the node allows reception of two different packets simultaneously or it can receive either of the classes but only one at a time.

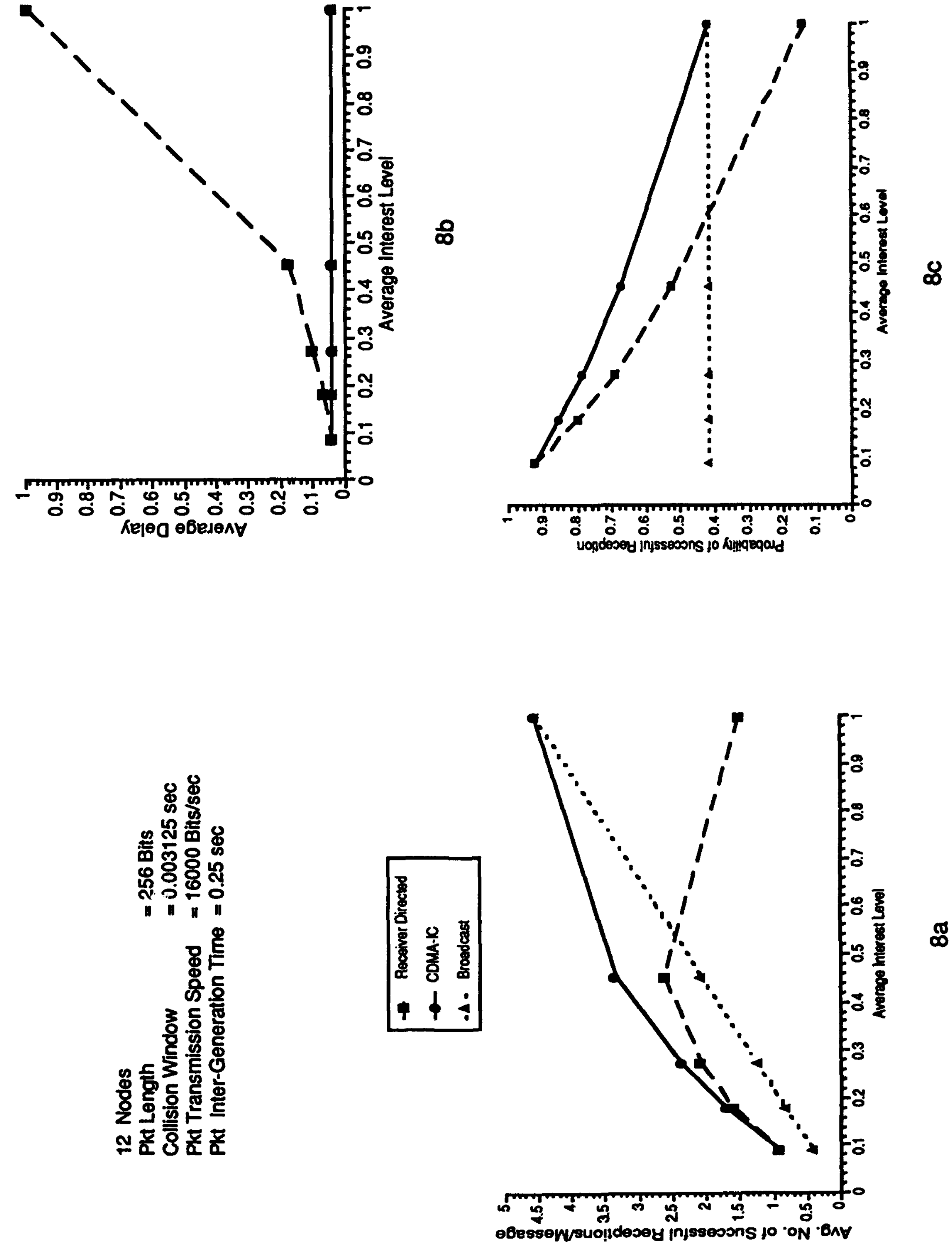
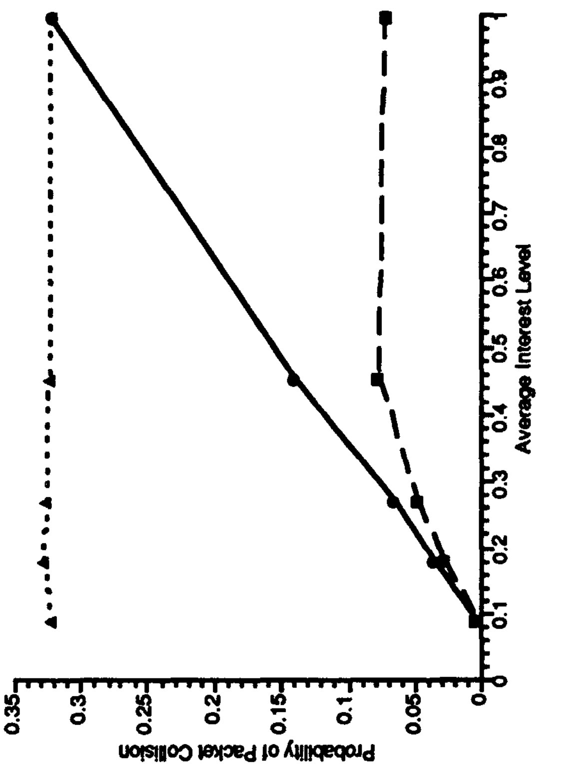
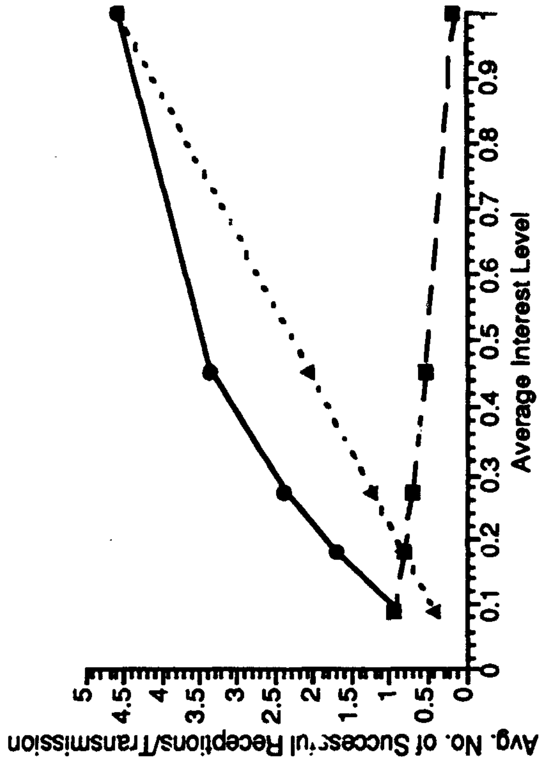


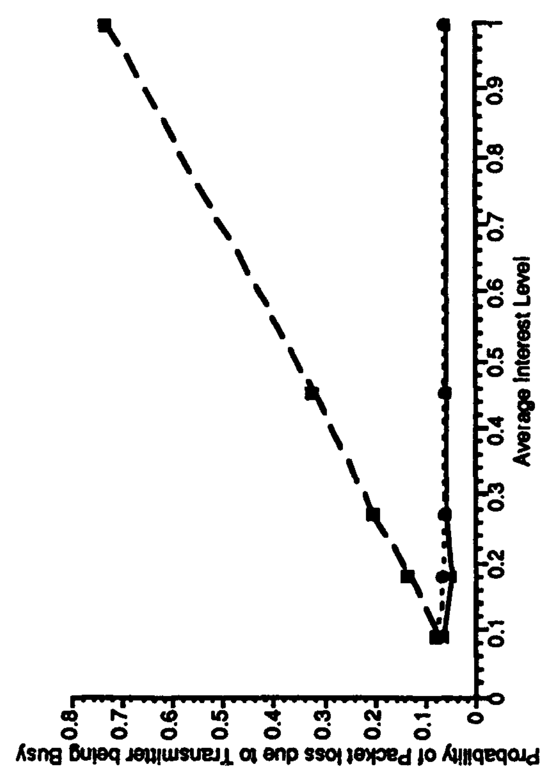
Figure 8. Comparison of CDMA-IC, CDMA-RD, and broadcasting - high traffic levels


Probability of Packet Collision
Average Interest Level

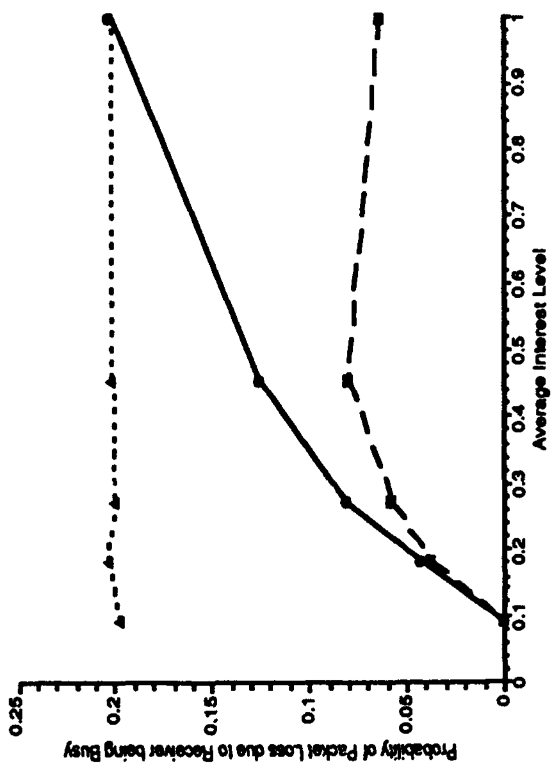
8f


Avg. No. of Successful Receptions/Transmission
Average Interest Level

8g


Probability of Packet loss due to Transmitter being Busy
Average Interest Level

8d


Probability of Packet Loss due to Receiver being Busy
Average Interest Level

8e

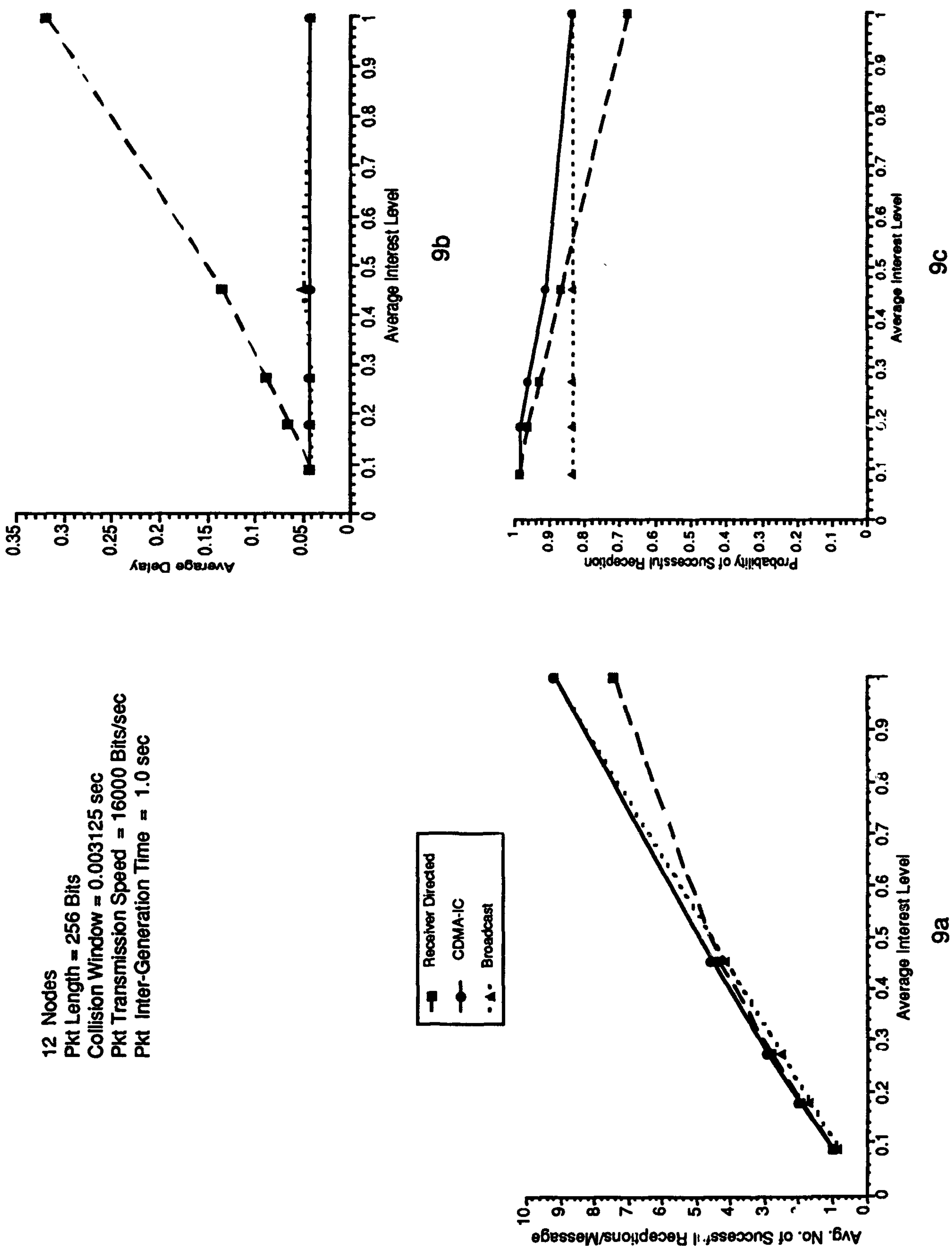
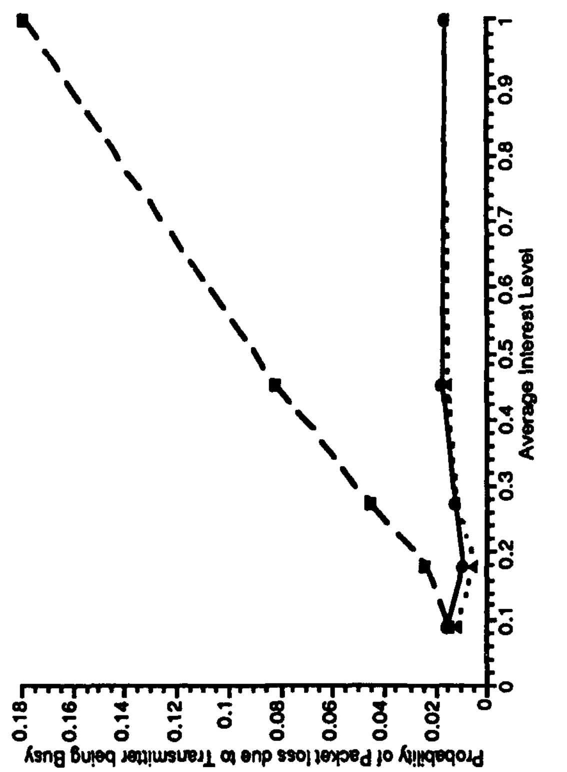
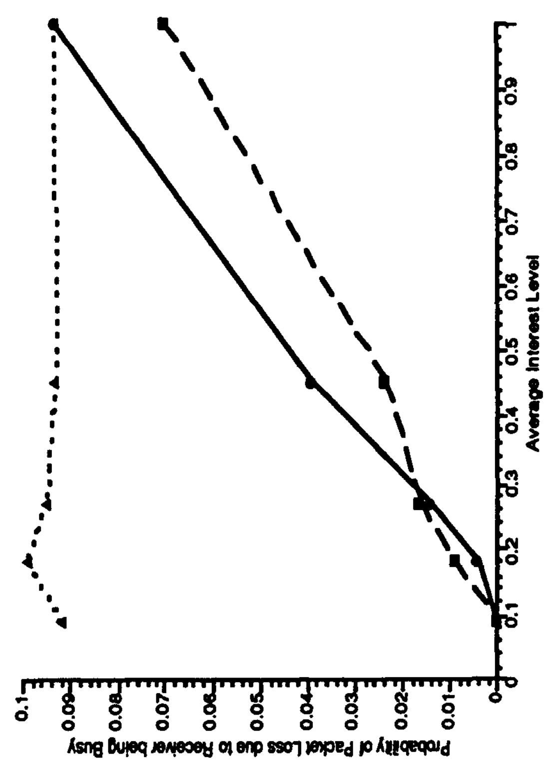


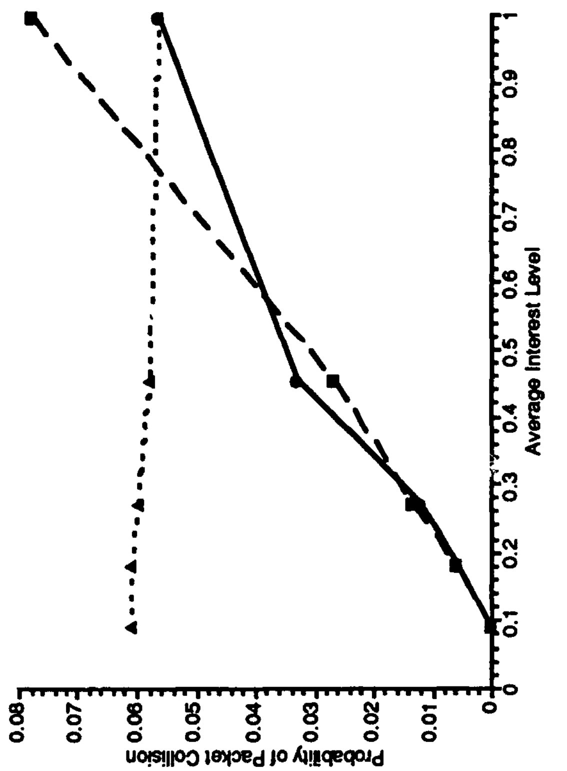
Figure 9. Comparison of CDMA-IC, CDMA-RD, and broadcasting - low traffic levels


Probability of Packet loss due to Transmitter being Busy
Average Interest Level

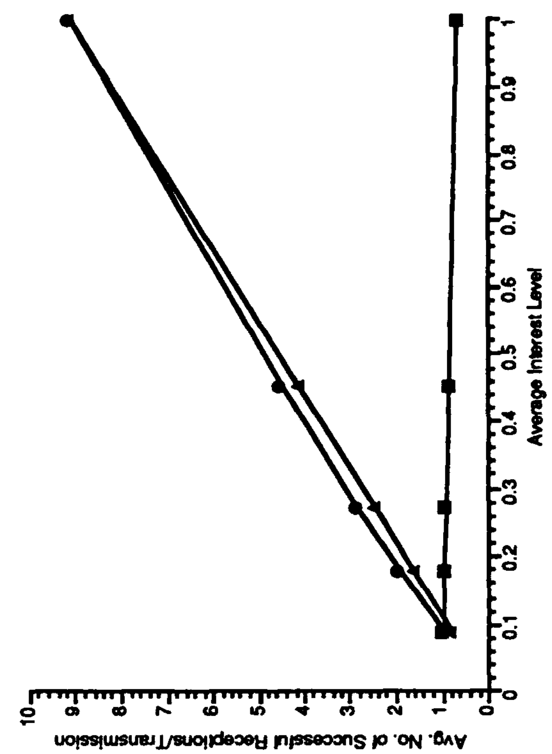
9d


Probability of Packet Loss due to Receiver being Busy
Average Interest Level

9e


Probability of Packet Collision
Average Interest Level

9f


Avg. No. of Successful Receptions/Transmission
Average Interest Level

9g

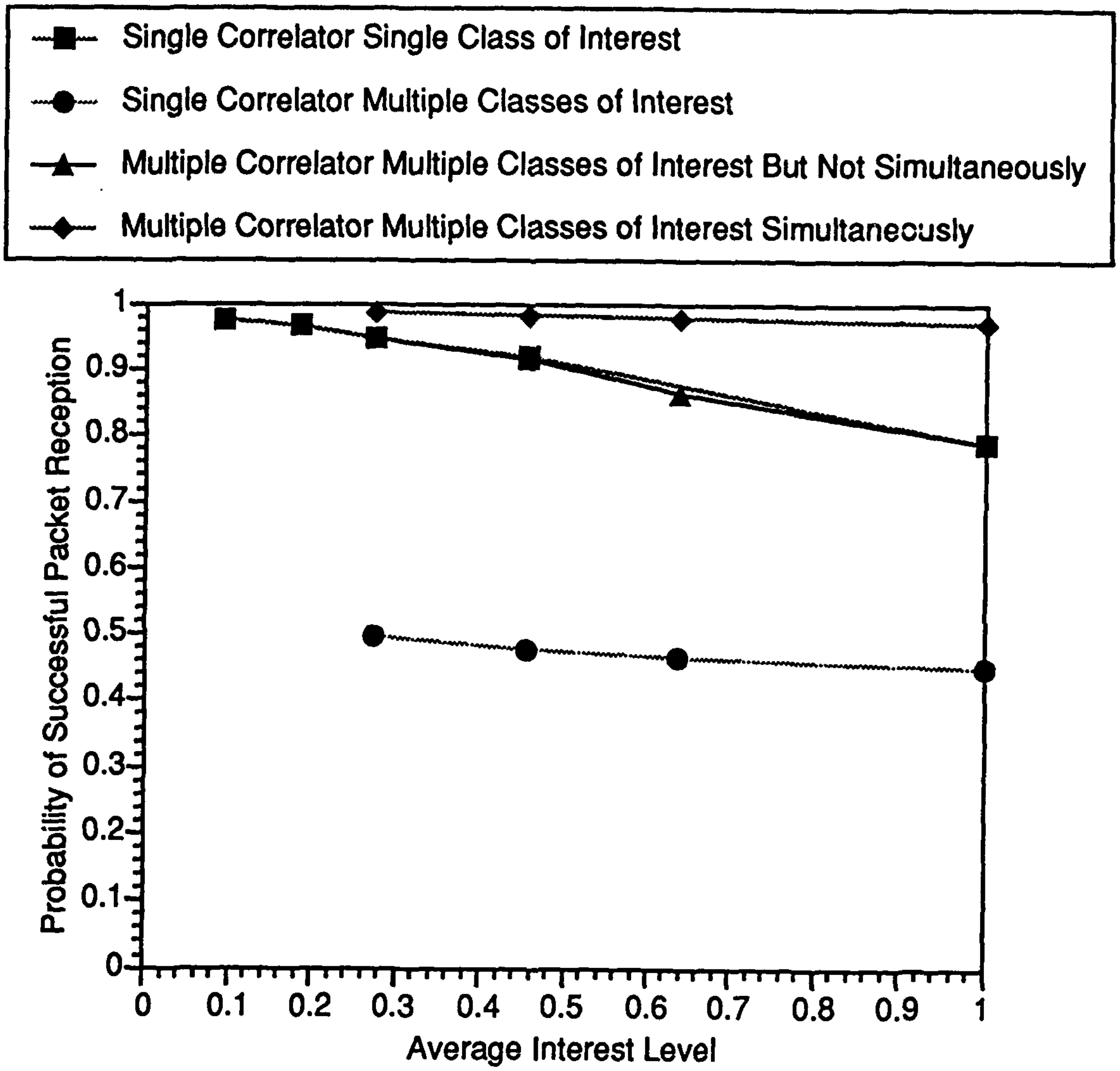


Figure 10. CDMA-IC: Multiple correlators - probability of successful packet reception

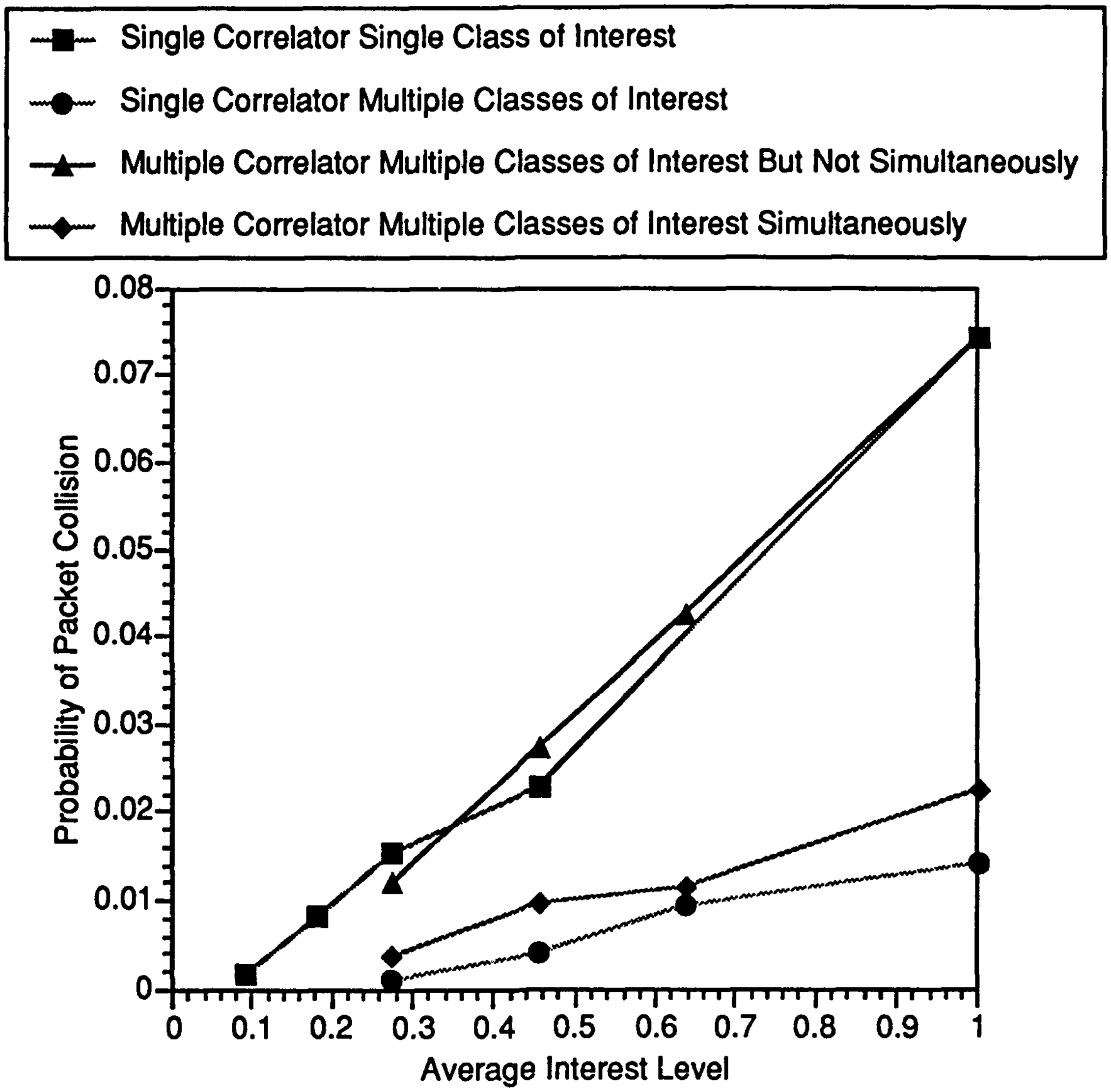


Figure 11. CDMA-IC: Multiple correlators - probability of packet collision

6. Concluding Remarks

We have proposed a new method of code allocation in Code Division Multiple Access, based on Information Classes. We have shown that the performance of this approach can exceed that of both receiver-directed codes or broadcast codes in environments where there are several multicast groups (in which a given message is of interest to more than one node), or where it is difficult to know what types of messages (out of a fixed set of message types) a receiver is interested in at any particular time. Performance can be further enhanced when receivers can simultaneously attempt to correlate multiple spreading codes, using multi-signal detection techniques.

There are a number of directions in which the concept needs to be further expanded and analyzed, such as development of general theoretical models for CDMA-IC schemes considering multiple correlators, dynamic class partitions and code assignments for non-homogeneous traffic scenarios, and extensions to multi-hop spread spectrum networks. In future work, we will investigate the performance of CDMA-IC in environments with bursty and time-critical traffic,when the receivers have multiple correlators to receive messages of multiple classes, and in environments where nodes frequently change state. Application of multiple detection theories that envisage weighting and combined processing of the outputs of the multiple correlators might be very useful in reducing the average error probability and combating near-far problems when power control is not feasible or not effective. We will also consider more detailed models of the transmitter's knowledge of the receivers' interests.

References

[1] J. E. Baker, I. Shahnawaz and A.R.K. Sastry, "Code Division Multiple Access with Code Assignments Using Information Classes," IEEE Military Communications Conference (MILCOM), pp. 556-560, Boston, October 1993.

[2]. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread Spectrum Communications," IEEE Trans. on Communications (Part I, Sp. issue on spread-spectrum communications), pp 855-884, May 1982.

[3] R. E. Ziemer and R. L. Peterson, *Digital Communications and Spread Spectrum Systems*, New York, New York: Macmillan, 1985.

[4] D. L. Nicholson, *Spread Spectrum Signal Design: L.P.E. and A.J. Systems,* Rockville, Maryland: Computer Science Press, 1988.

[5]. D. L. Schilling, R. L. Pickholtz, and L. B. Milstein (Guest Editors), *Spread Spectrum Communications I and II*, IEEE Journal on Selected Areas on Communications, Special issues, May and June 1990.

[6]. K.L. Kosbar (Guest Editor), *Spread Spectrum*, IEEE Journal on Selected Areas on Communications, Special issue, May 1992.

[7]. R. A, Scholtz, "Optimal CDMA codes," Conf. Rec., IEEE Nat. Telecomm. Conf., pp. 54.2.1-54.2.4, 1979.

[8]. R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing," IEEE Trans. on Inform. Theory, pp. 619-621, October 1967.

[9]. S. Davidovici, L. B. Milstein, and D. L. Schilling, "A New Rapid Acquisition Technique for Direct Sequence Spread Spectrum Communications," IEEE Trans. on Commun., pp. 1161-1168, November 1984.

[10]. J. A. Rajan, "Adaptive Acqusition of Multiple Access Codes," Conf. Rec., IEEE Int. Conf. on Commun., pp 7E.1.1-7E.1.4, June 1982.

[11]. R. E. Kahn, S. A. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in Packet Radio Technology," Proc. IEEE, vol. 66, pp 1468-1496, Nov. 1978.

[12]. C. L. Weber, G. K. Huth and B. H. Batson, "Performance Considerations of Code Division Multiple Access Systems," IEEE Trans. on Veh. Tech. vol. VT-30, pp. 3-9, February 1981.

[13]. D. Raychaudhuri, "Performance Analysis of Random Access Packet Switched Code Division Multiple Access Systems," IEEE Trans. on Commun., vol. COM-29, pp. 895-901, June 1981.

[14]. J. M. Musser and J. N. Daigle, "Throughput Analysis of an Asynchronous Code Division Multiple Access (CDMA) System," in Conf. Rec., IEEE Int. Conf. on Commun., pp. 2F-2-1-2F-2-7, June 1982.

[15]. P. Economopoulos and M. L. Molle, "On the performance of slotted ALOHA in a Spread spectrum Environment," Symposium Record, ACM SIGCOMM Symposium on Communication Architectures and Protocols, pp. 234- -241, June 1984.

[16]. F. A. Tobagi, `Multiaccess Protocols in Packet Communication Systems," IEEE Trans. on Commun., pp 468-488, April 1980.

[17]. M. Molle and L. Kleinrock, "Virtual Time CSMA: Why two clocks are better than one," IEEE Trans. on Commun.,. vol. COM-33, no. 9, pp. 919-933, September 1985.

[18]. M. L. Molle, "Asynchronous multiple access tree algorithms," Symp. Record, ACM SIGCOMM'83,. pp. 214-218, March 1983.

[19]. J. I. Capetanakis, "Tree algorithms for packet broadcast channels," IEEE Trans. on Inform. Theory, pp. 505-515, Sept. 1979.

[20]. J. Kurose, M. Schwartz, and Y. Yemini, "Controlling window Protocols for Time-Constrained Communications in Multiple Access Networks," IEEE Trans. on Commun.,. vol. COM-36, no. 1, pp. 41-49, January 1988.

[21]. D. P. Gerakoulis, T. N. Saadawi, and D. L. Schilling, "A Class of Tree Algorithms with Variable Message Length," Symposium Record, ACM SIGCOMM Symposium on Communication Architectures and Protocols, pp. 242-247, June 1984.

[22]. J. L. Massey, " Collision-resolution algorithms and random-access communications," UCLA Tech. Rep. no. UCLA-ENG-8016, April 1980. Also in Multi-User Commun., Ed. G. Longo, Springer-Verlag CISM Courses and Lecture Series.

[23]. D. Ryter, "A conflict resolution algorithm for noisy multiaccess channels," B.S. thesis, EECS Dept., Mass. Inst. Tech., May 1980.

[24]. T. T. Liu and D. Towsley, "Window and tree protocols for satellite channels," Proc. INFOCOM 83., pp. 215-221, April 1983.

[25]. M. J. Ferguson, "A Bound and Approximation of Delay Distribution for Fixed Length Packets in an Unslotted ALOHA Channel and a Comparison with Time Division Multiplexing (TDM)," IEEE Trans. on Commun., pp. 136-139, January 1977.

[26]. M. J. Ferguson, "An Approximate Analysis of Delay for Fixed and Variable Length Packets in an Unslotted ALOHA channel," IEEE Trans. on Commun., pp. 644-654, July 1977.

[27]. S. Bellini and F. Borgonovo, "On the Throughput of an ALOHA Channel with variable length Packets," IEEE Trans. on Commun., vol. COM-28, pp. 1932-1935, Nov. 1980.

[28]. L. Kleinrock and S. S. Lam, "Packet switching in a multiaccess broadcast channel: Performance evaluation," IEEE Trans. on Commun., pp. 410-423, April 1975.

[29]. L. P. Clare, "Control procedures for slotted ALOHA systems that achieve stability," Proc. ACM SIGCOMM '86 Symposium., pp. 302-309, Stowe, Vermont, August 1986.

[30]. K. Joseph and D. Raychaudhuri, "Stability analysis of asynchronous random access CDMA systems," Proc. IEEE GLOBECOM'86,. pp. 48.2.1-48.2.7, Dec. 1986.

[31]. B. Hajek, "Recursive retransmission control - Application to a frequency-hopped spread-spectrum system," Proc. Conf. Infor. Sciences and Systems,. pp. 116-120, March 1982.

[32]. E. Geraniotis and R. Mani, "Throughput analysis of random access tree protocol for frequency-hopped spread-spectrum packet radio networks," IEEE MILCOM'87,. pp. 23.7.1-23.7.6, October 1987.

[33]. E. Geraniotis and R. Mani, "Throughput analysis of random access tree protocol for direct-sequence spread-spectrum packet radio networks," Proc. Conf. on Decision and Control,. pp. 687-692, December 1987.

[34]. E. Sousa and J. Silvester, "A spreading code protocol for a distributed spread spectrum packet radio network," Proc. IEEE GLOBECOM'84,. pp.481-486, Nov. 1984.

[35]. E. Sousa and J. Silvester, "A code switching technique for distributed spread spectrum packet radio networks," Proc. IEEE Int. Conf. on Commun., June 1985.

[36] L.P. Clare, J.E. Baker, and A.R.K. Sastry, "A Performance Comparison of Control Policies for Slotted ALOHA Frequency-Hopped Systems," Conf. Record, pp. 608-614, IEEE MILCOM'90, Monterey, October 1-3, 1990.

[37]. L. P. Clare and A. R. K. Sastry, "The effects of slotting, burstiness, and jamming in frequency-hopped random access systems," Conf. Rec, IEEE MILCOM, pp. 154-160, 1989.

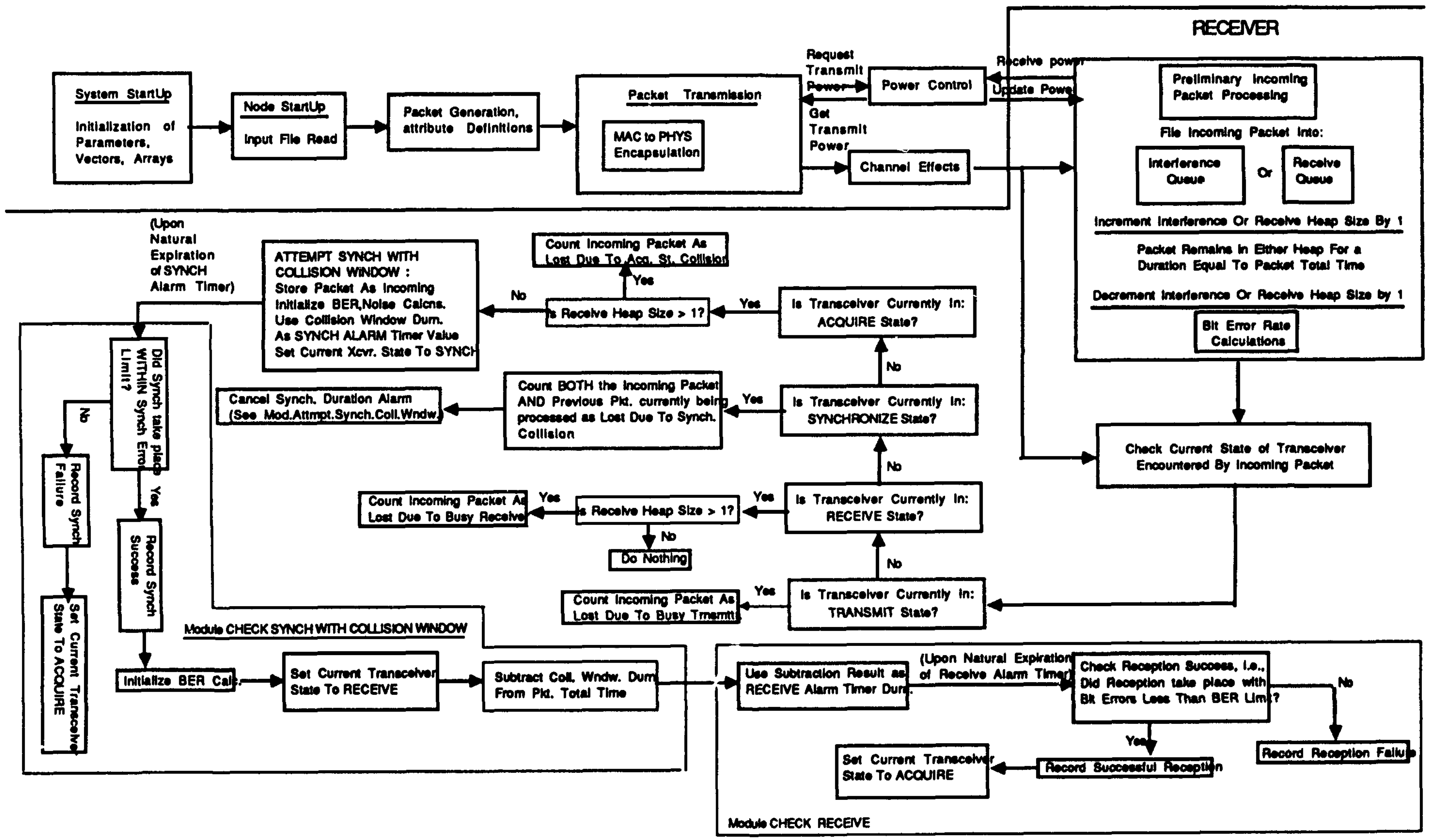
[38]. R. L. Cruz, "Protocols for multiaccess channels with continuous entry and noisy feedback," Laboratory for Information and Decision Systems, Mass. Inst. Tech., Cambridge, MA, Rep. LIDS-TH-1213, May 1982.

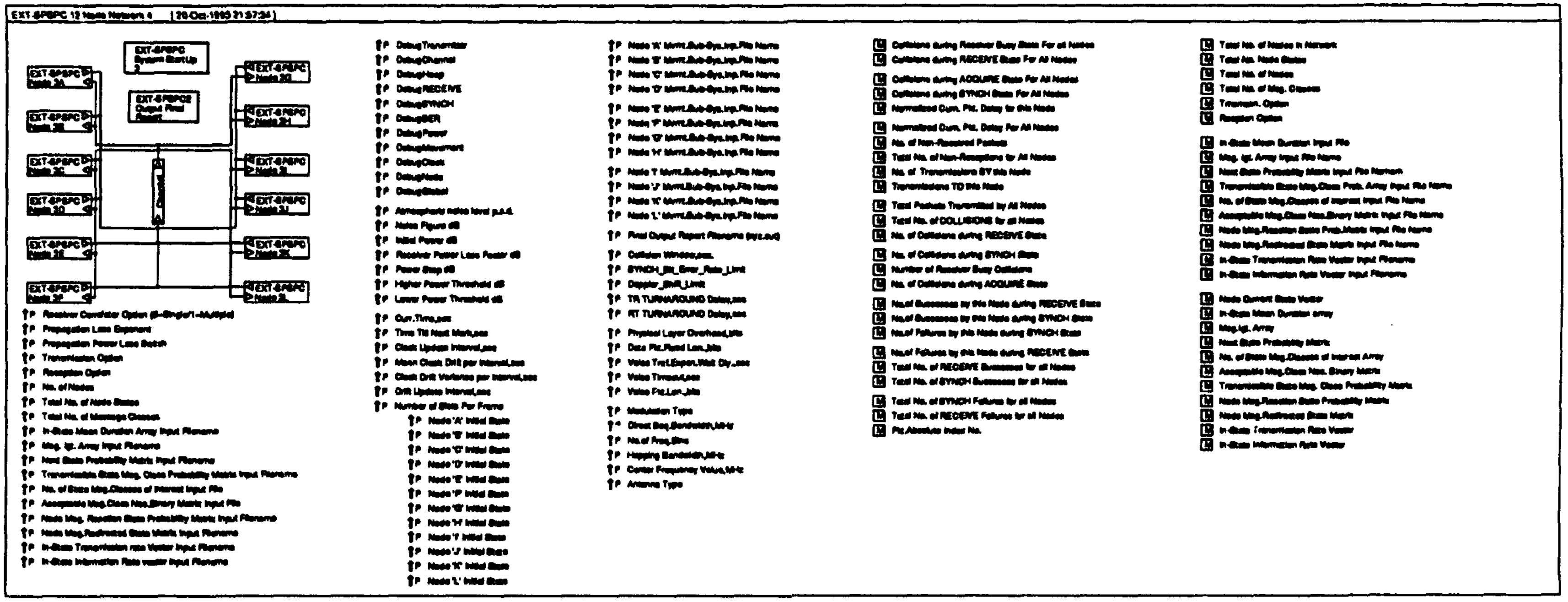
[39] E. S. Sousa, ``Interference modeling in a direct-sequence spread-spectrum packet radio network," IEEE Transactions on Communications}, vol. 38, pp. 1475--1482, September 1990.

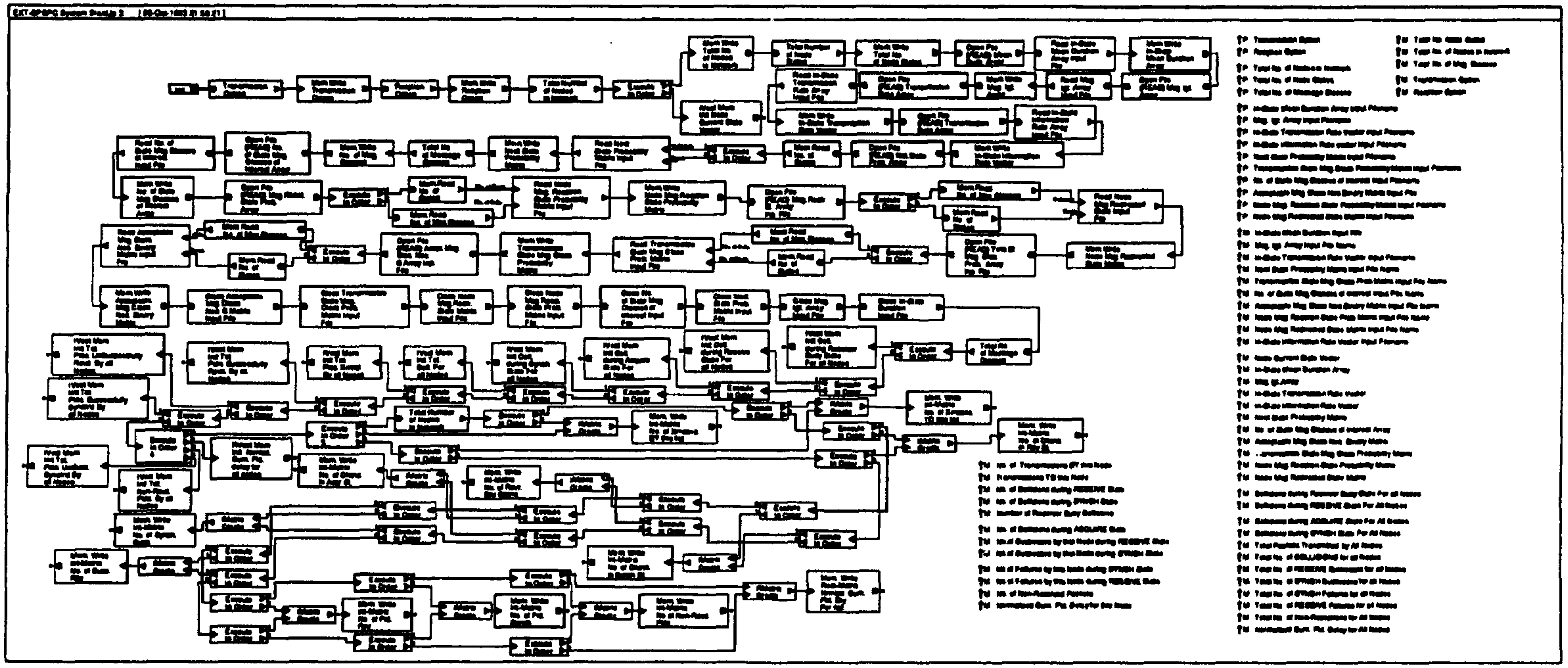
[40]. Comdisco Systems, Inc., *Block Oriented Network Simulator (BONeS™) User's Guide*, 1990.

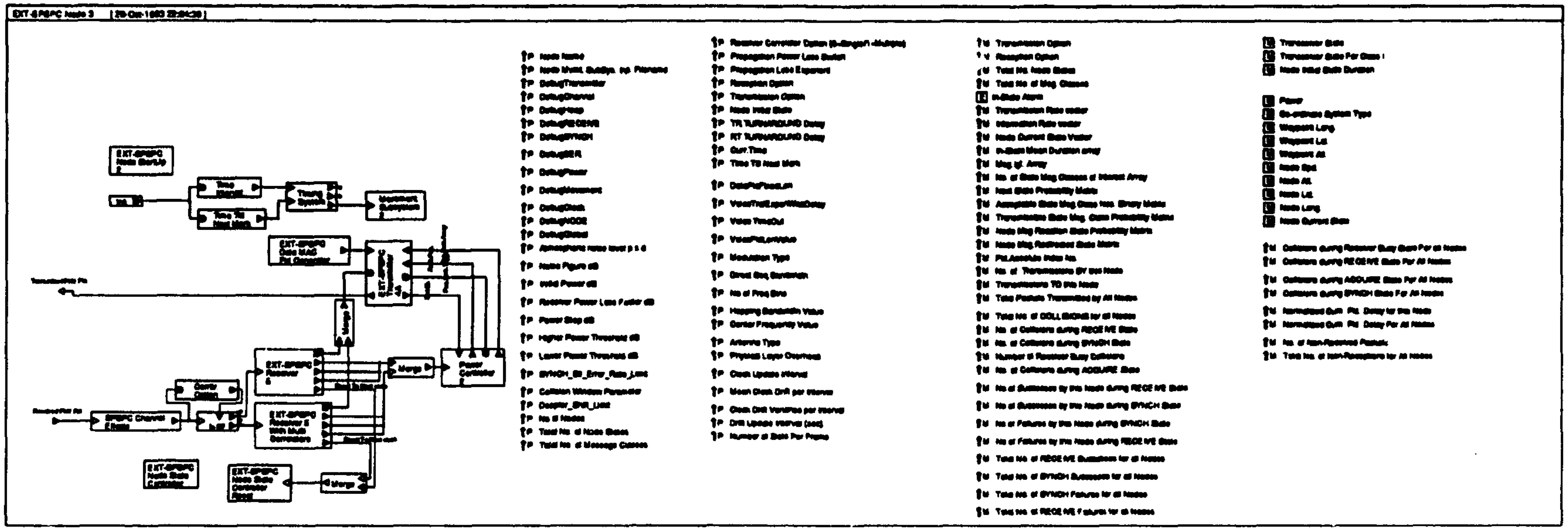
APPENDIX: SIMULATION BLOCK DIAGRAMS

A STREAMLINED OVERALL SYSTEM BLOCK DIAGRAM FOR THE SPREAD SPECTRUM NETWORK



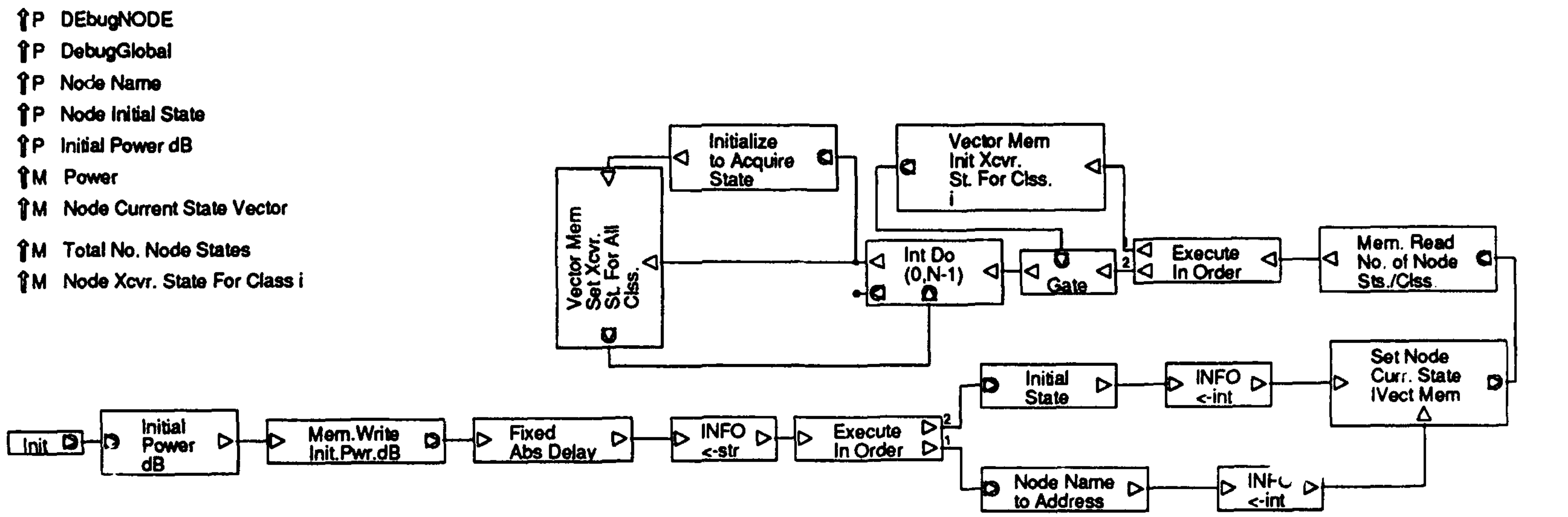

EXT-SPSPC 12 Node Network 4 [29-Oct-1993 21:57:34]
EXT-SPSPC System Start Up 2
EXT-SPSPC2 Output Final Report
EXT-SPSPC Node 3A
EXT-SPSPC Node 3B
EXT-SPSPC Node 3C
EXT-SPSPC Node 3D
EXT-SPSPC Node 3E
EXT-SPSPC Node 3F
EXT-SPSPC Node 3G
EXT-SPSPC Node 3H
EXT-SPSPC Node 3I
EXT-SPSPC Node 3J
EXT-SPSPC Node 3K
EXT-SPSPC Node 3L
Channel
Receiver Correlator Option (0=Single/1=Multiple)
Propagation Loss Exponent
Propagation Power Loss Switch
Transmission Option
Reception Option
No. of Nodes
Total No. of Node States
Total No. of Message Classes
In-State Mean Duration Array Input Filename
Msg. Igt. Array Input Filename
Next State Probability Matrix Input Filename
Transmittable State Msg. Class Probability Matrix Input Filename
No. of State Msg.Classes of Interest Input File
Acceptable Msg.Class Nos.Binary Matrix Input File
Node Msg. Reaction State Probability Matrix Input Filename
Node Msg.Redirected State Matrix Input Filename
In-State Transmission rate Vector Input Filename
In-State Information Rate vector Input Filename
DebugTransmitter
DebugChannel
DebugHeap
DebugRECEIVE
DebugSYNCH
DebugBER
DebugPower
DebugMovement
DebugClock
DebugNode
DebugGlobal
Atmospheric noise level p.s.d.
Noise Figure dB
Initial Power dB
Receiver Power Loss Factor dB
Power Step dB
Higher Power Threshold dB
Lower Power Threshold dB
Curr.Time,sec
Time Till Next Mark,sec
Clock Update Interval,sec
Mean Clock Drift per Interval,sec
Clock Drift Variance per Interval,sec
Drift Update Interval,sec
Number of Slots Per Frame
Node 'A' Initial State
Node 'B' Initial State
Node 'C' Initial State
Node 'D' Initial State
Node 'E' Initial State
Node 'F' Initial State
Node 'G' Initial State
Node 'H' Initial State
Node 'I' Initial State
Node 'J' Initial State
Node 'K' Initial State
Node 'L' Initial State
Node 'A' Msmt.Sub-Sys.Inp.File Name
Node 'B' Msmt.Sub-Sys.Inp.File Name
Node 'C' Msmt.Sub-Sys.Inp.File Name
Node 'D' Msmt.Sub-Sys.Inp.File Name
Node 'E' Msmt.Sub-Sys.Inp.File Name
Node 'F' Msmt.Sub-Sys.Inp.File Name
Node 'G' Msmt.Sub-Sys.Inp.File Name
Node 'H' Msmt.Sub-Sys.Inp.File Name
Node 'I' Msmt.Sub-Sys.Inp.File Name
Node 'J' Msmt.Sub-Sys.Inp.File Name
Node 'K' Msmt.Sub-Sys.Inp.File Name
Node 'L' Msmt.Sub-Sys.Inp.File Name
Final Output Report Filename (xyz.out)
Collision Window,sec
SYNCH_Bit_Error_Rate_Limit
Doppler_Shift_Limit
TR TURNAROUND Delay,sec
RT TURNAROUND Delay,sec
Physical Layer Overhead,bits
Data Pkt.Fixed Len.,bits
Voice Trst.Expon.Wait Dly.,sec
Voice Timeout,sec
Voice Pkt.Len.,bits
Modulation Type
Direct Seq.Bandwidth,MHz
No.of Freq.Bins
Hopping Bandwidth,MHz
Center Frequency Value,MHz
Antenna Type
Collisions during Receiver Busy State For all Nodes
Collisions during RECEIVE State For All Nodes
Collisions during ACQUIRE State For All Nodes
Collisions during SYNCH State For All Nodes
Normalized Cum. Pkt. Delay for this Node
Normalized Cum. Pkt. Delay For All Nodes
No. of Non-Received Packets
Total No. of Non-Receptions for All Nodes
No. of Transmissions BY this Node
Transmissions TO this Node
Total Packets Transmitted by All Nodes
Total No. of COLLISIONS for all Nodes
No. of Collisions during RECEIVE State
No. of Collisions during SYNCH State
Number of Receiver Busy Collisions
No. of Collisions during ACQUIRE State
No.of Successes by this Node during RECEIVE State
No.of Successes by this Node during SYNCH State
No.of Failures by this Node during SYNCH State
No.of Failures by this Node during RECEIVE State
Total No. of RECEIVE Successes for all Nodes
Total No. of SYNCH Successes for all Nodes
Total No. of SYNCH Failures for all Nodes
Total No. of RECEIVE Failures for all Nodes
Pkt.Absolute Index No.
Total No. of Nodes in Network
Total No. Node States
Total No. of Nodes
Total No. of Msg. Classes
Transmsn. Option
Reception Option
In-State Mean Duration Input File
Msg. Igt. Array Input File Name
Next State Probability Matrix Input File Namem
Transmittable State Msg.Class Prob. Array Input File Name
No. of State Msg.Classes of Interest Input File Name
Acceptable Msg.Class Nos.Binary Matrix Input File Name
Node Msg.Reaction State Prob.Matrix Input File Name
Node Msg.Redirected State Matrix Input File Name
In-State Transmission Rate Vector Input Filename
In-State Information Rate Vector Input Filename
Node Current State Vector
In-State Mean Duration array
Msg.Igt. Array
Next State Probability Matrix
No. of State Msg.Classes of Interest Array
Acceptable Msg.Class Nos. Binary Matrix
Transmittable State Msg. Class Probability Matrix
Node Msg.Reaction State Probability Matrix
Node Msg.Redirected State Matrix
In-State Transmission Rate Vector
In-State Information Rate Vector

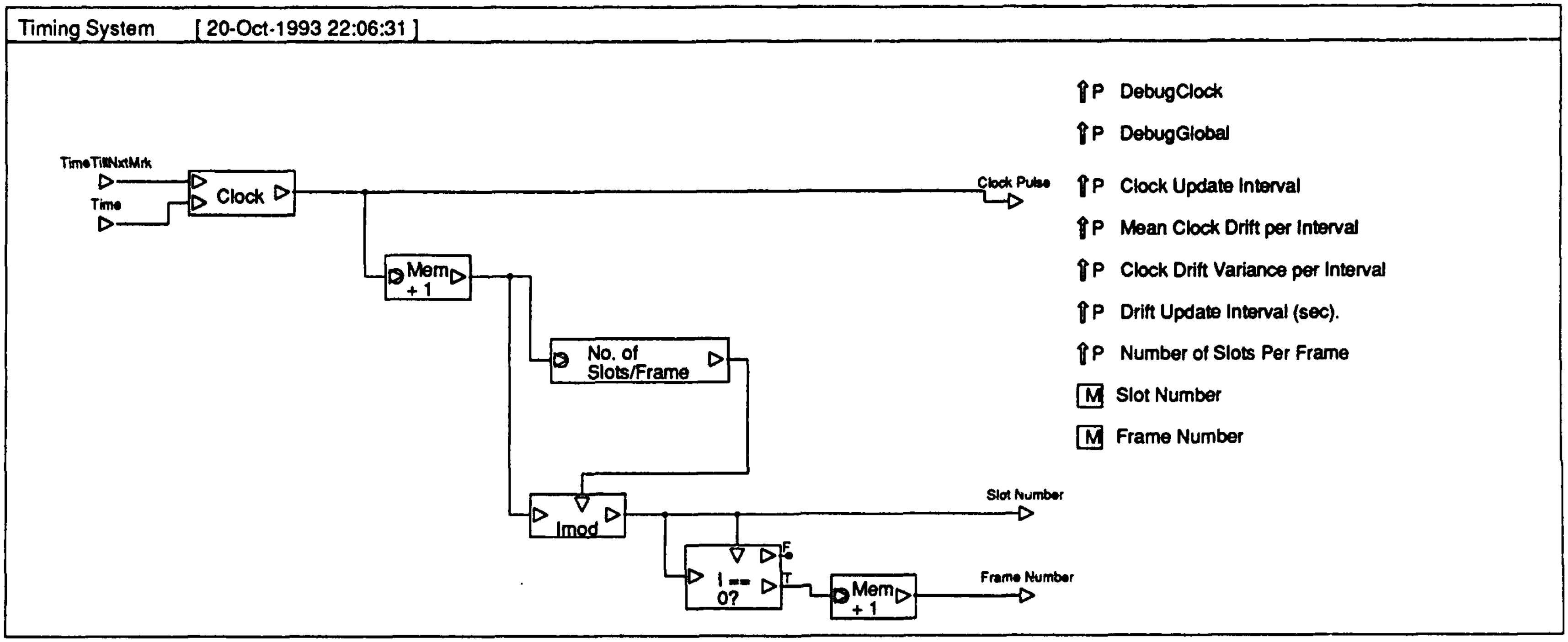


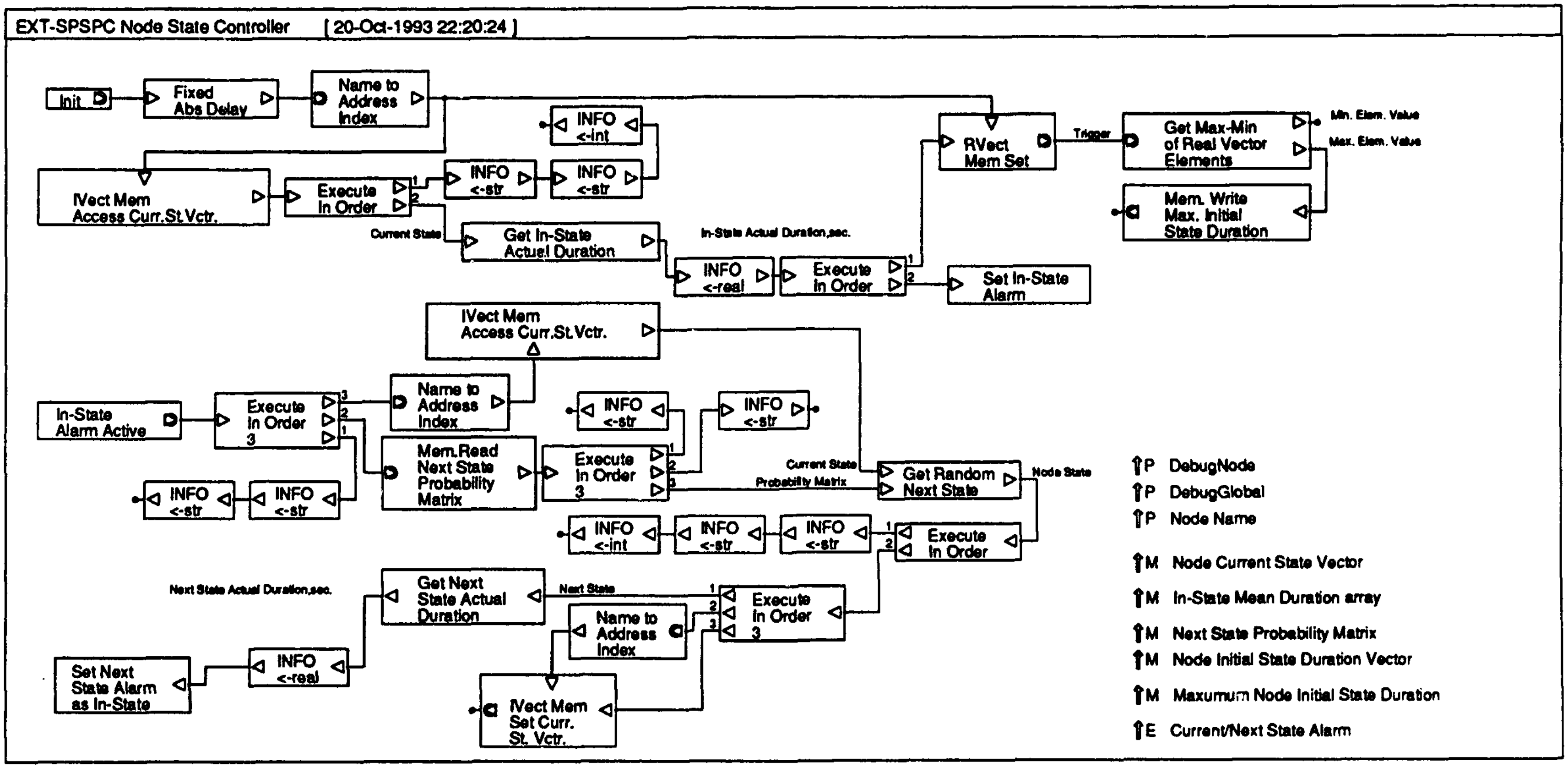


EXT-SPSPC
Merge
Power Controller
Node Name
DebugTransmitter
DebugChannel
DebugRECEIVE
DebugSYNCH
DebugBER
DebugPower
DebugMovement
DebugClock
DebugNODE
DebugGlobal
Noise Figure dB
Power Step dB
Higher Power Threshold dB
Lower Power Threshold dB
SYNCH_Bit_Error_Rate_Limit
Collision Window Parameter
Doppler_Shift_Limit
No of Nodes
Total No. of Node States
Total No. of Message Classes
Propagation Power Loss Switch
Propagation Loss Exponent
Reception Option
Transmission Option
Node Initial State
TR TURNAROUND Delay
RT TURNAROUND Delay
Curr Time
Time Til Next Mark
Video TimeOut
Modulation Type
No of Freq Bins
Hopping Bandwidth Value
Center Frequency Value
Antenna Type
Physical Layer Overhead
Clock Update Interval
Mean Clock Drift per Interval
Drift Update Interval (sec)
Number of Slots Per Frame
Transmission Option
Reception Option
Total No. Node States
Total No. of Msg. Classes
In-State Alarm
Transmission Rate vector
Information Rate vector
Node Current State Vector
Transmissions TO this Node
Total Packets Transmitted by All Nodes
Number of Receiver Busy Collisions
Transceiver State
Node Initial State Duration
Power
Co-ordinate System Type
Waypoint Long
Waypoint Lat
Waypoint Alt
Node Spd
Node Alt
Node Lat
Node Long
Node Current State

EXT-SPSPC Node StartUp 2 [20-Oct-1993 22:05:49]

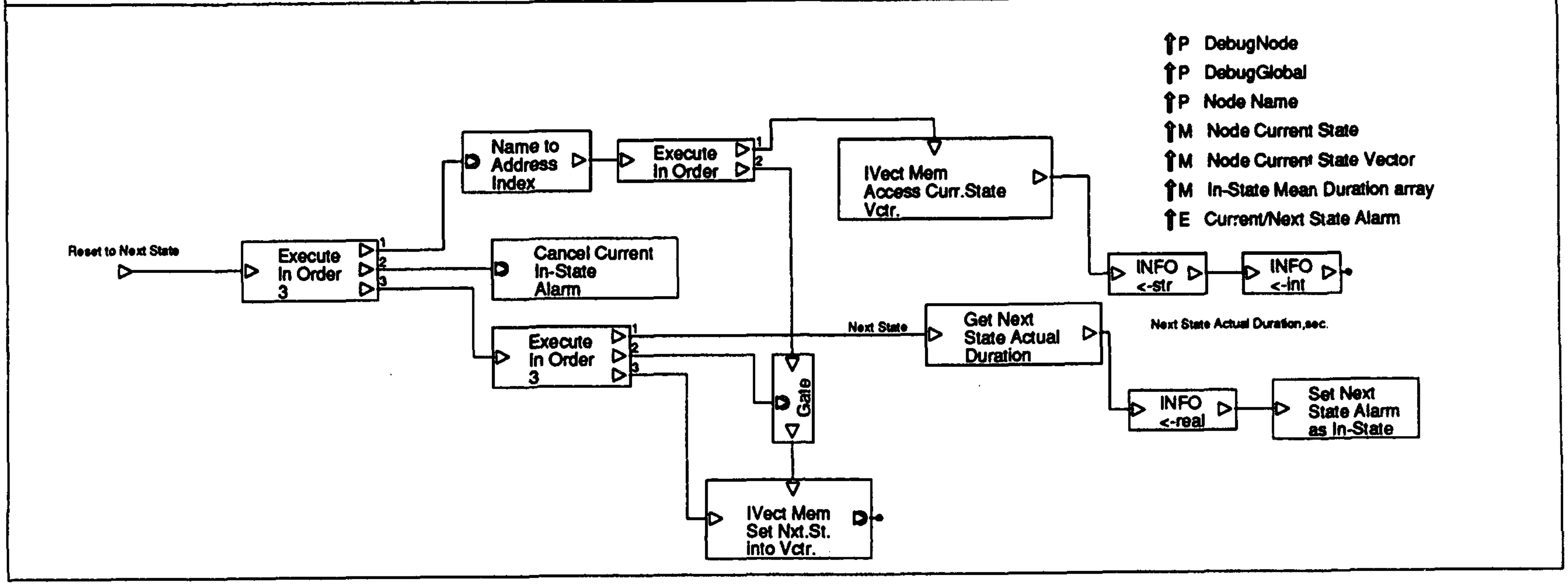


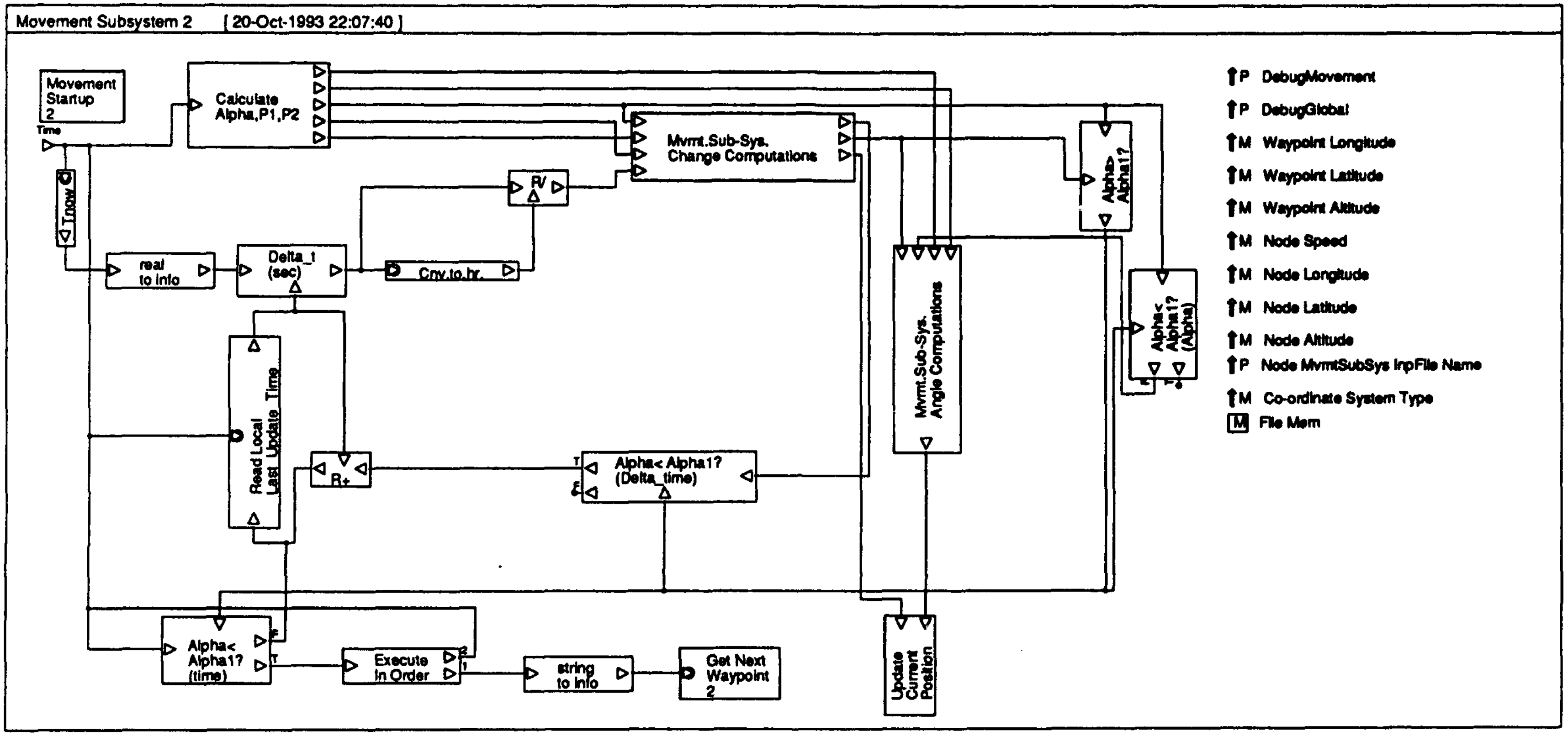

Timing System
[20-Oct-1993 22:06:31]
TimeTillNxtMrk
Time
Clock
Clock Pulse
Mem
+ 1
No. of
Slots/Frame
Imod
Slot Number
| ==
0?
F
T
Mem
+ 1
Frame Number
P DebugClock
P DebugGlobal
P Clock Update Interval
P Mean Clock Drift per Interval
P Clock Drift Variance per Interval
P Drift Update Interval (sec).
P Number of Slots Per Frame
M Slot Number
M Frame Number



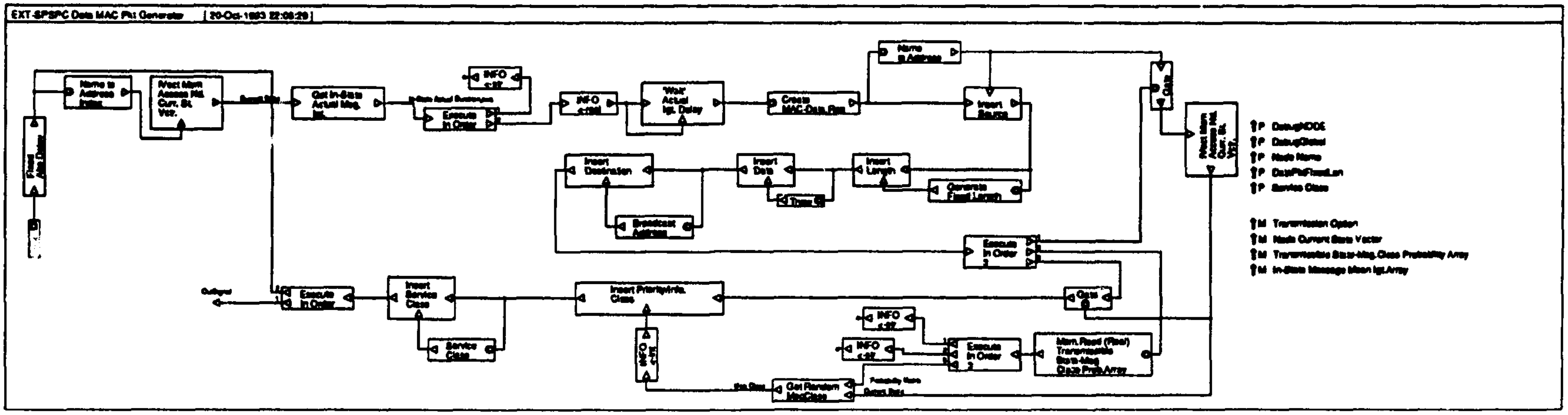
EXT-SPSPC Node State Controller [20-Oct-1993 22:20:24]
Init
Fixed Abs Delay
Name to Address Index
IVect Mem Access Curr.St.Vctr.
Execute In Order
INFO <-int
INFO <-str
INFO <-str
Current State
Get In-State Actual Duration
In-State Actual Duration,sec.
INFO <-real
Execute In Order
Set In-State Alarm
RVect Mem Set
Trigger
Get Max-Min of Real Vector Elements
Min. Elem. Value
Max. Elem. Value
Mem. Write Max. Initial State Duration
IVect Mem Access Curr.St.Vctr.
In-State Alarm Active
Execute In Order 3
Name to Address Index
Mem.Read Next State Probability Matrix
INFO <-str
INFO <-str
INFO <-str
INFO <-str
Execute In Order 3
Current State
Probability Matrix
Get Random Next State
Node State
INFO <-int
INFO <-str
INFO <-str
Execute In Order
Next State Actual Duration,sec.
Get Next State Actual Duration
Next State
Execute In Order 3
Name to Address Index
Set Next State Alarm as In-State
INFO <-real
IVect Mem Set Curr. St. Vctr.
P DebugNode
P DebugGlobal
P Node Name
M Node Current State Vector
M In-State Mean Duration array
M Next State Probability Matrix
M Node Initial State Duration Vector
M Maxumum Node Initial State Duration
E Current/Next State Alarm

EXT-SPSPC Node State Controller Reset [20-Oct-1993 22:19:05]

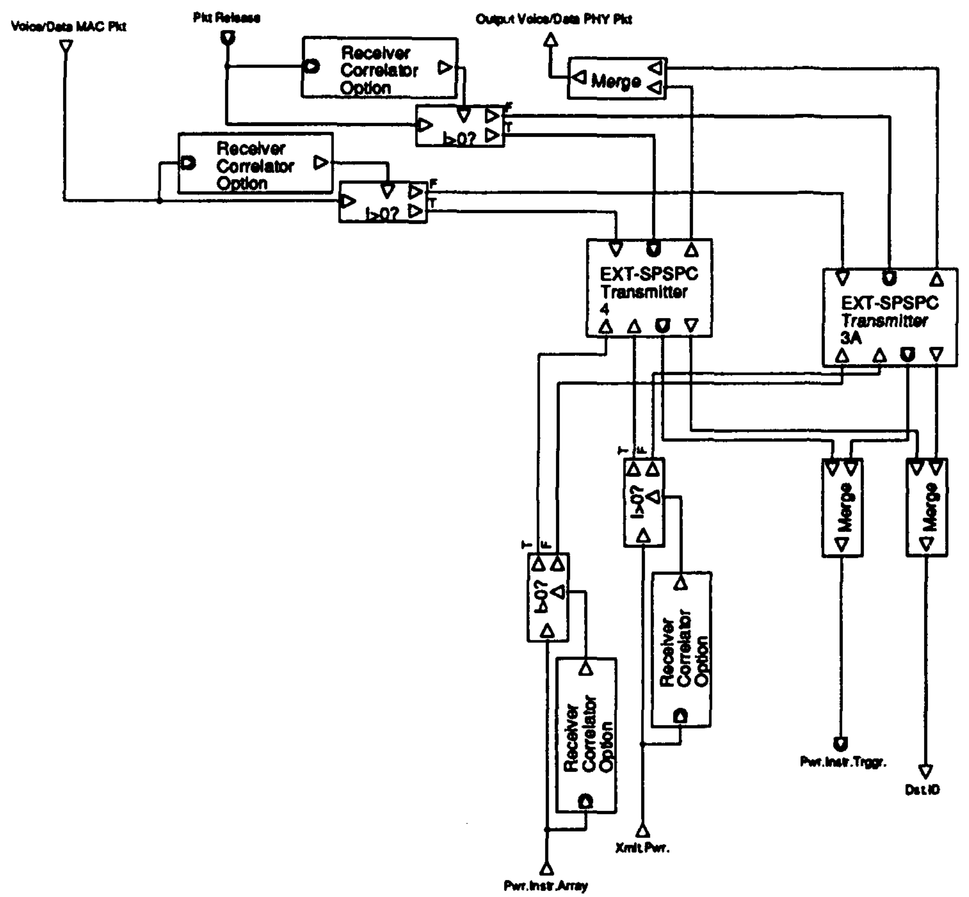




Movement Subsystem 2
[20-Oct-1993 22:07:40]
Movement Startup 2
Time
Tnow
Calculate Alpha,P1,P2
real to info
Delta_t (sec)
Cnv.to.hr.
R/
Mvmt.Sub-Sys. Change Computations
Read Local Last Update Time
R+
Alpha< Alpha1? (Delta_time)
T
F
Mvmt.Sub-Sys. Angle Computations
Alpha> Alpha1?
Alpha< Alpha1? (Alpha)
Alpha< Alpha1? (time)
Execute In Order
1
2
string to info
Get Next Waypoint 2
Update Current Position
P DebugMovement
P DebugGlobal
M Waypoint Longitude
M Waypoint Latitude
M Waypoint Altitude
M Node Speed
M Node Longitude
M Node Latitude
M Node Altitude
P Node MvmtSubSys InpFile Name
M Co-ordinate System Type
M File Mem


EXT-SPSPC Data MAC Pkt Generator [20-Oct-1993 22:08:29]
Fixed Attr Delay
Name to Address Index
Get In-State Actual Msg. lgt.
Execute In Order
INFO
'Wait' Actual lgt. Delay
Create MAC-Data_Pkt
Name to Address
Insert Source
Gate
Insert Destination
Broadcast Address
Insert Data
Insert Length
Generate Fixed Length
Execute In Order 2
Insert Service Class
Service Class
Insert Priority Class
Get Random MsgClass
Execute In Order 2
INFO
OutSignal
P DebugMODE
P DebugGlobal
P Node Name
P DataPktFixedLen
P Service Class
M Transmission Option
M Node Current State Vector
M Transmissible State-Msg.Class Probability Array
M In-State Message Mean lgt.Array

EXT-SPSPC Transmitter 4A [20-Oct-1993 22:09:06]

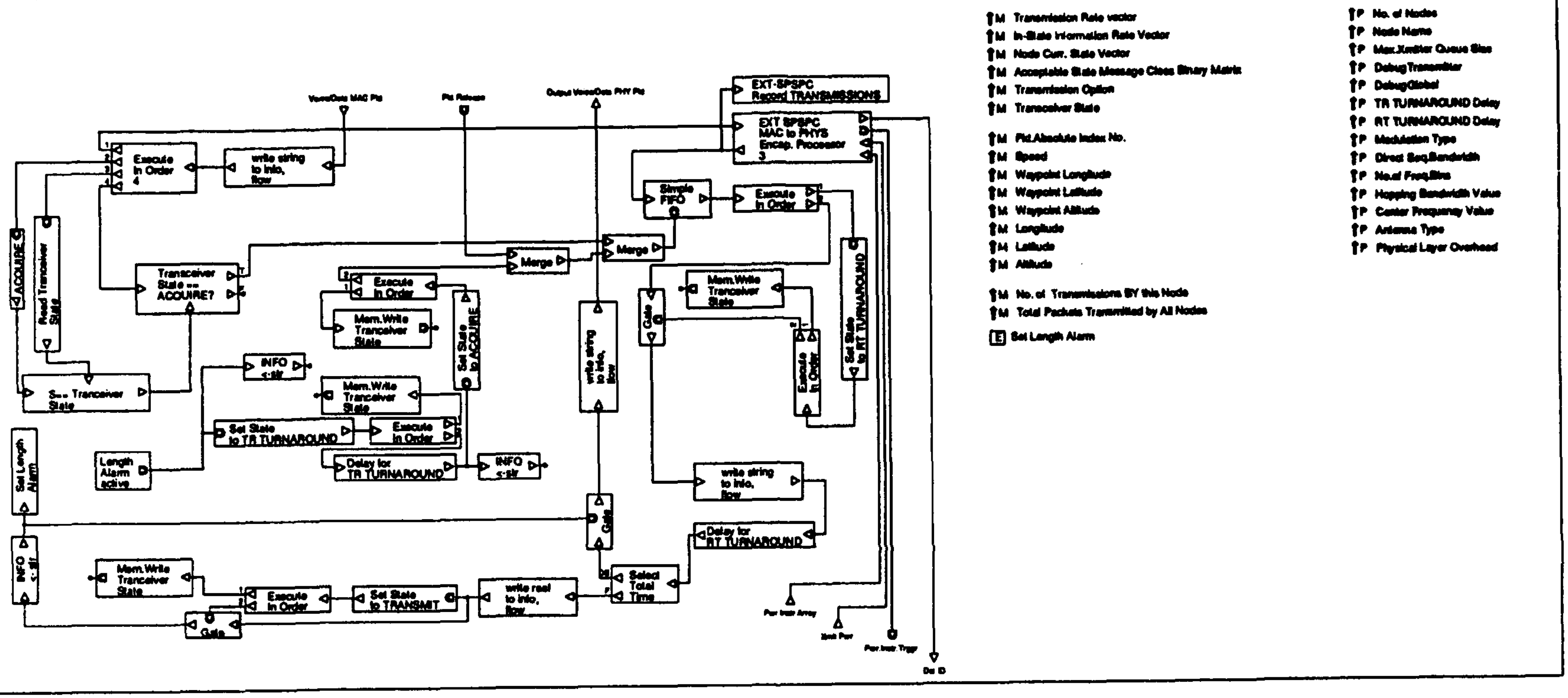


P Receiver Correlator Option (0=Single/1=Multiple)
P No. of Nodes
P Node Name
P Max.Xmitter Queue Size
P DebugTransmitter
P DebugGlobal
P Modulation Type
P Direct Seq.Bandwidth
P No.of Freq.Bins
P Hopping Bandwidth Value
P Center Frequency Value
P Antenna Type
P Physical Layer Overhead
P TR TURNAROUND Delay
P RT TURNAROUND Delay

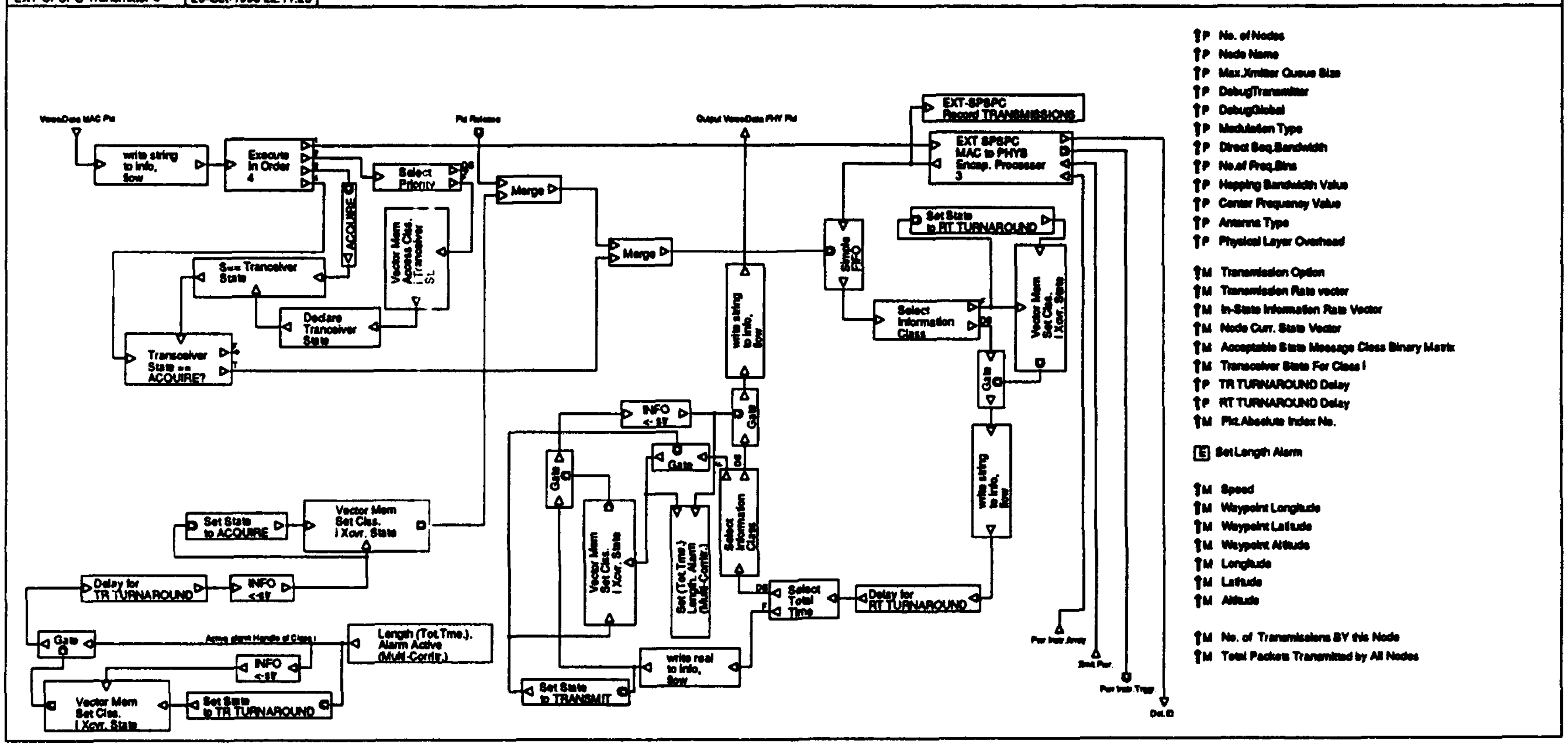
M Transmission Option
M Transmission Rate vector
M In-State Information Rate Vector
M Node Curr. State Vector
M Acceptable State Message Class Binary Matrix
M Transceiver State For Class I
M Transceiver State
M Pkt.Absolute Index No.
M Speed
M Waypoint Longitude
M Waypoint Latitude
M Waypoint Altitude
M Longitude
M Latitude
M Altitude

M No. of Transmissions BY this Node
M Total Packets Transmitted by All Nodes

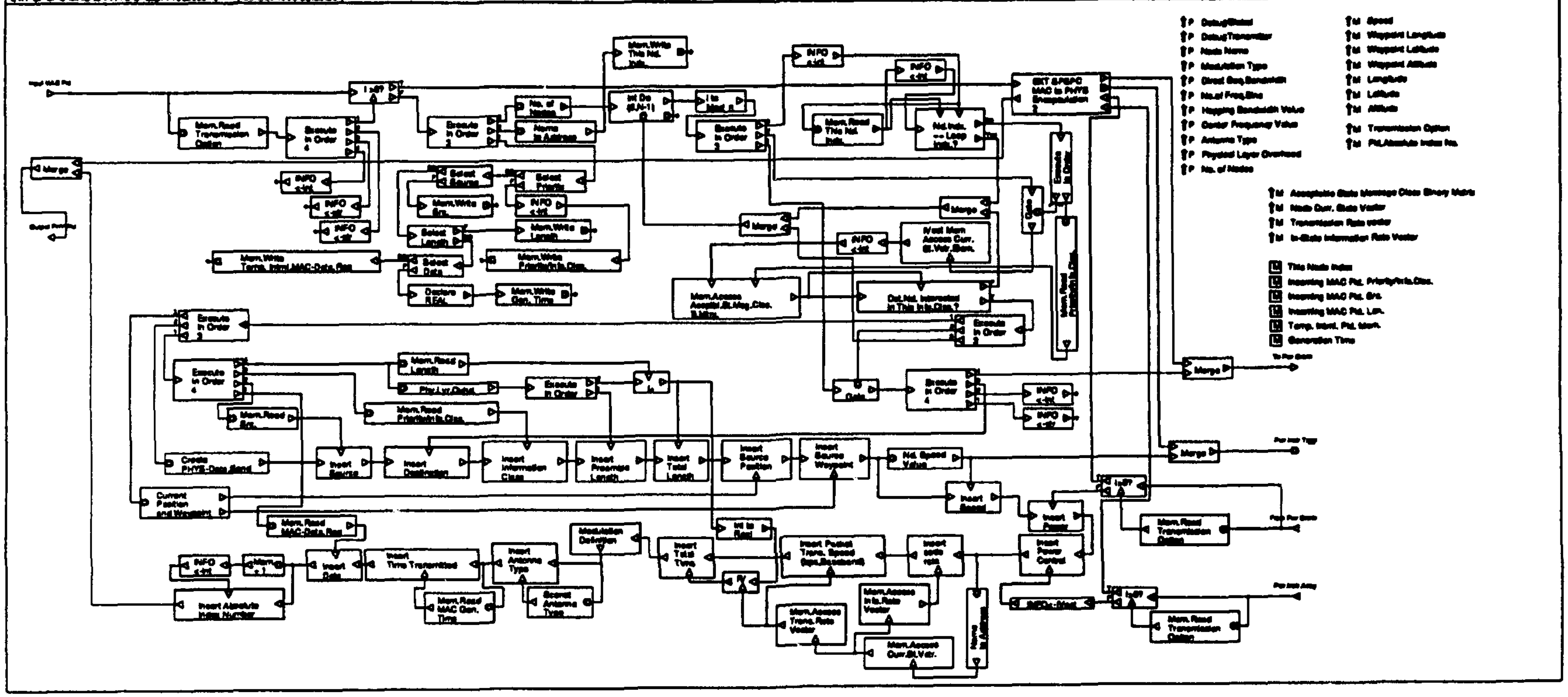
EXT-SPSPC Transmitter 3A [20-Oct-1993 22:10:38]

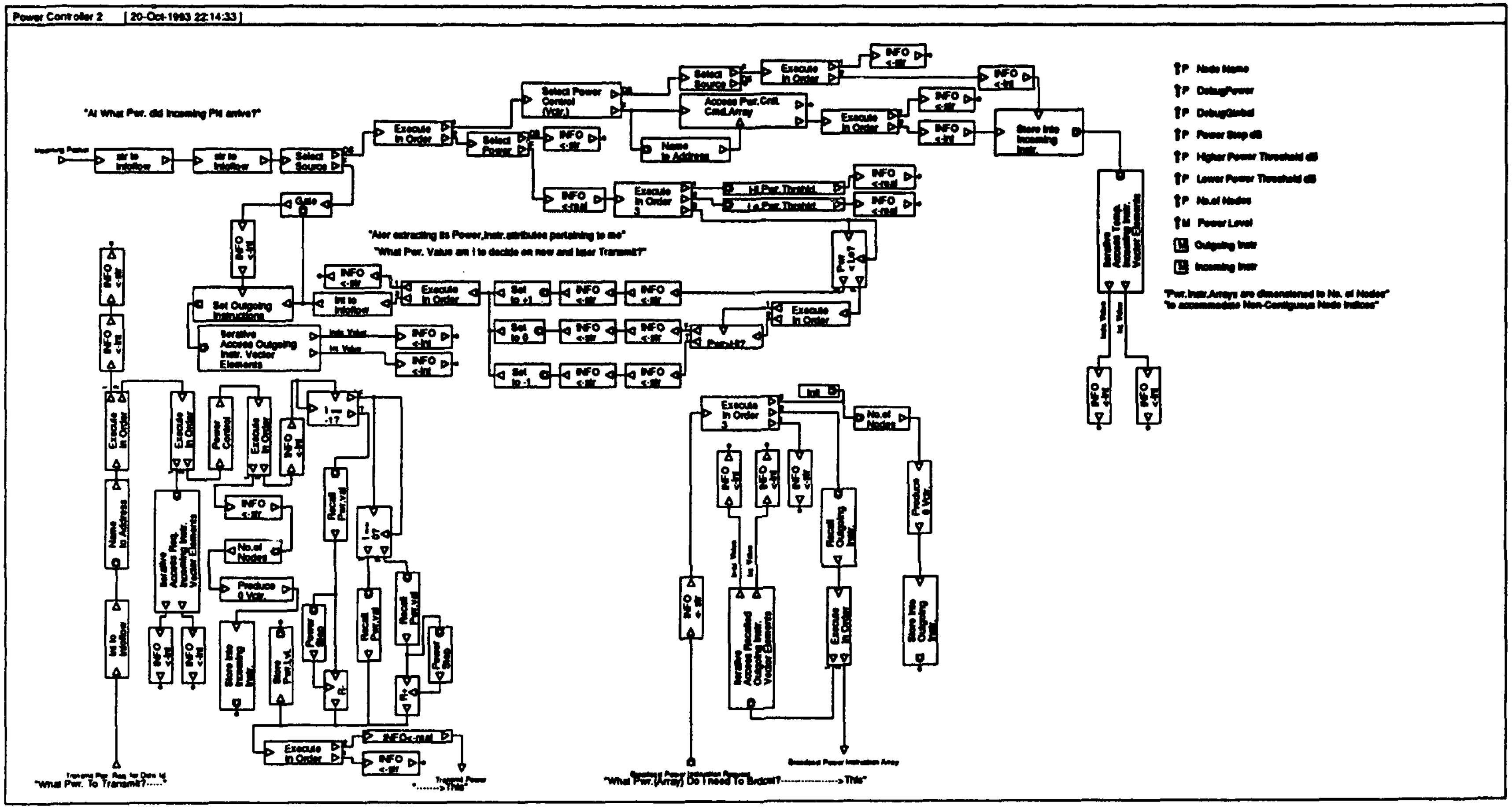


EXT-SPSPC Transmitter 4 [20-Oct-1993 22:11:25]

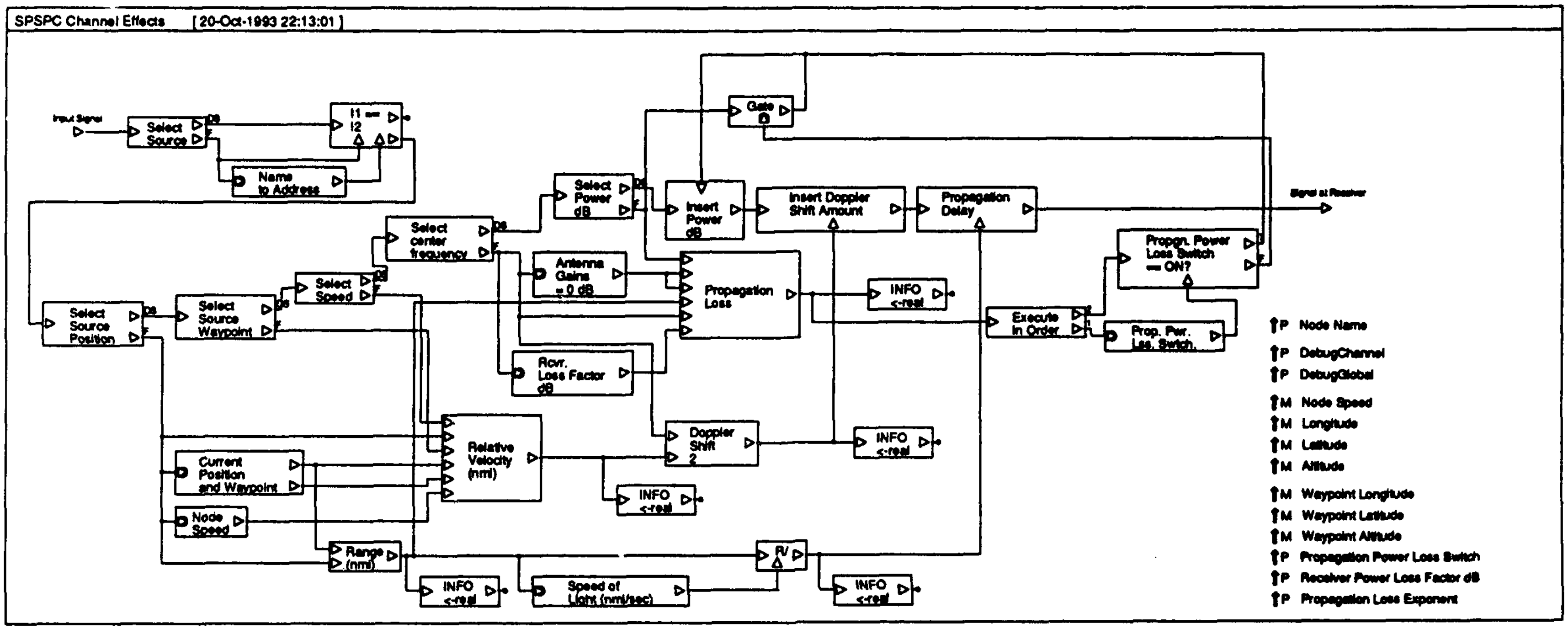


EXT SPSPC MAC to PHYS Encap. Processor 3 [20-Oct-1993 22:38:58]

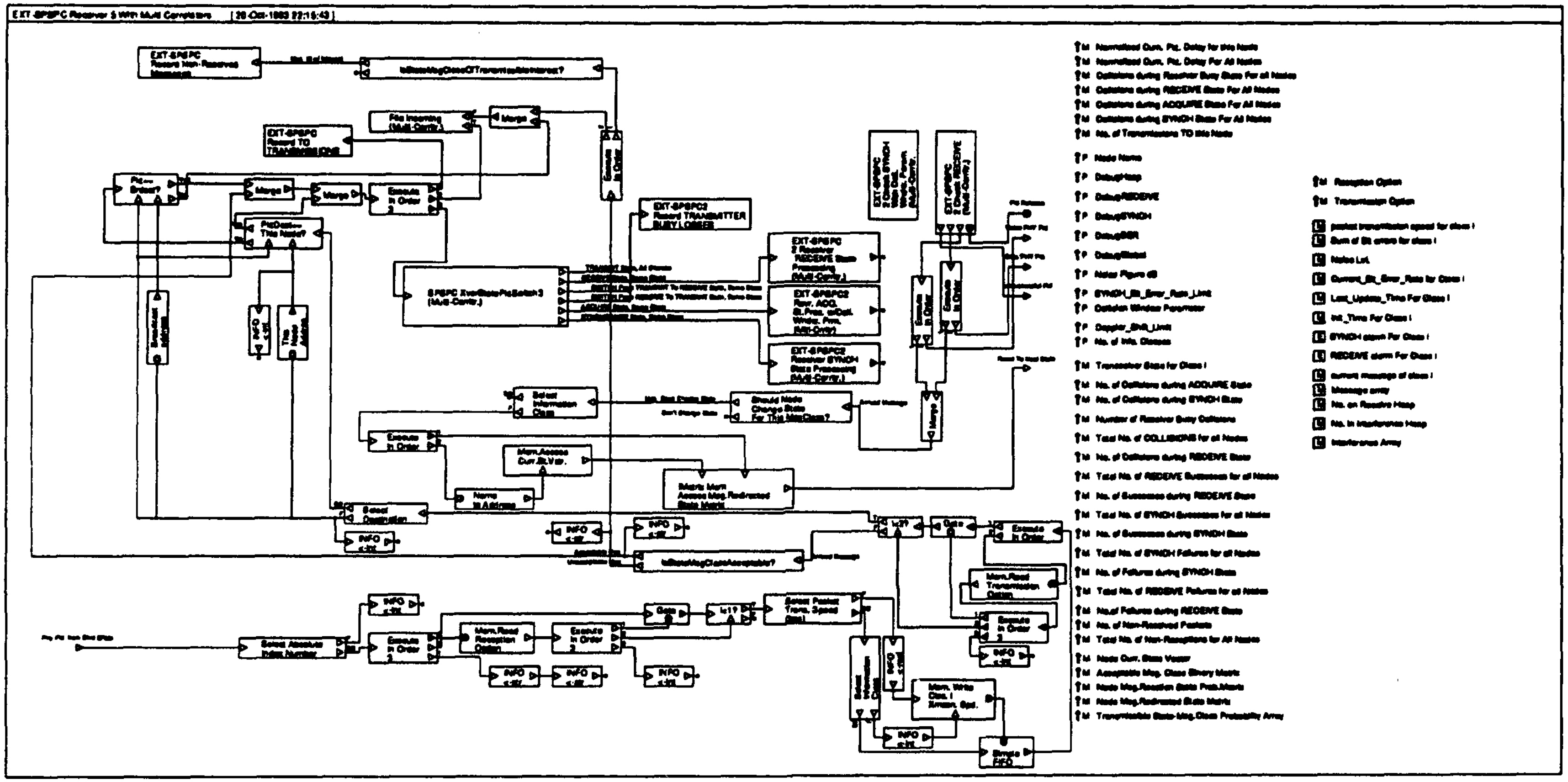




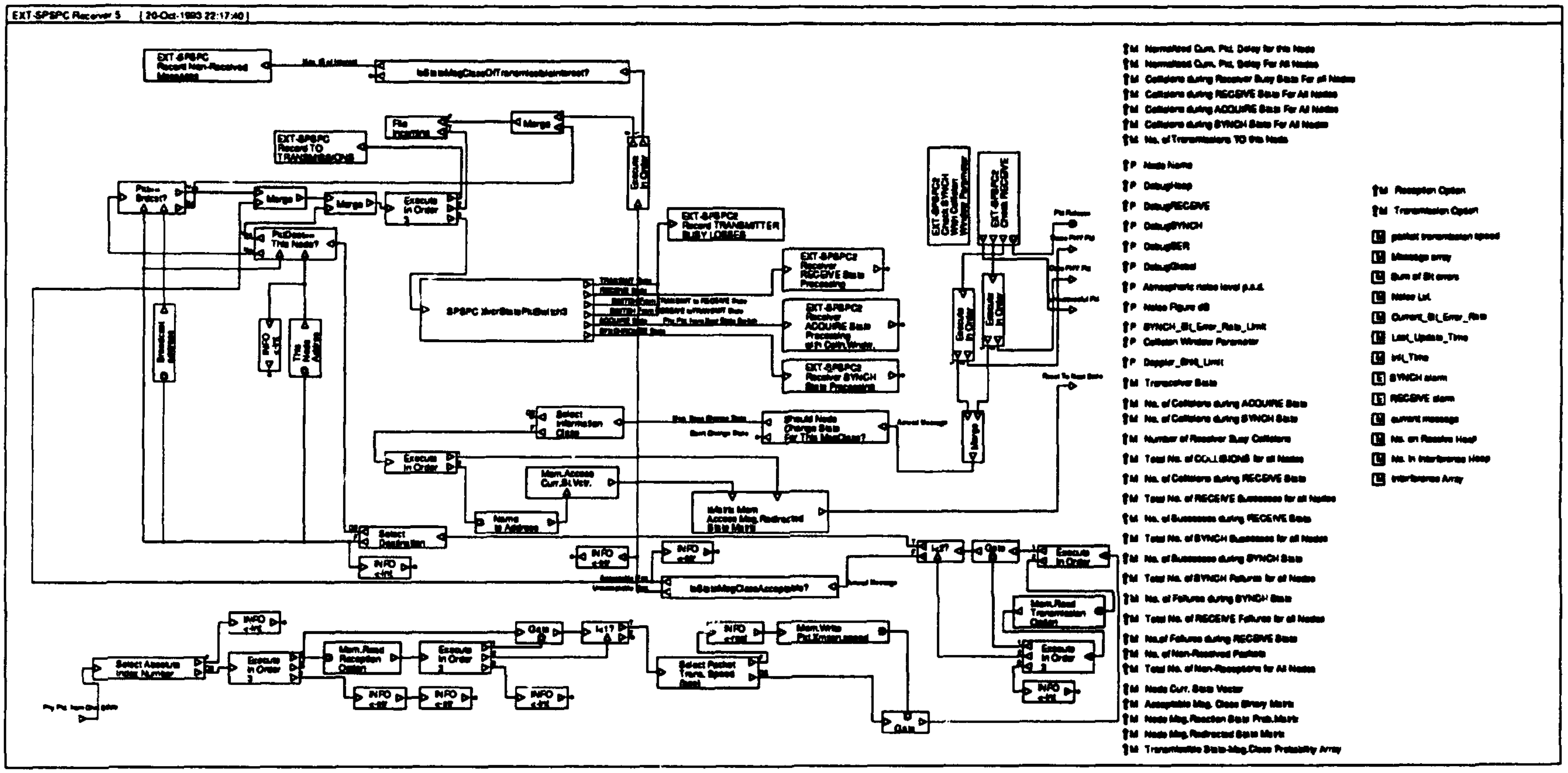
Power Controller 2
[20-Oct-1993 22:14:33]
"At What Pwr. did Incoming Pkt arrive?"
"After extracting its Power,Instr.attributes pertaining to me"
"What Pwr. Value am I to decide on now and later Transmit?"
"What Pwr. To Transmit?....."
"What Pwr.(Array) Do I need To Bcast?------------------> This"
Node Name
DebugPower
DebugGlobal
Power Step dB
Higher Power Threshold dB
Lower Power Threshold dB
No.of Nodes
Power Level
Outgoing Instr
Incoming Instr
"Pwr.Instr.Arrays are dimensioned to No. of Nodes"
"to accommodate Non-Contiguous Node Indices"



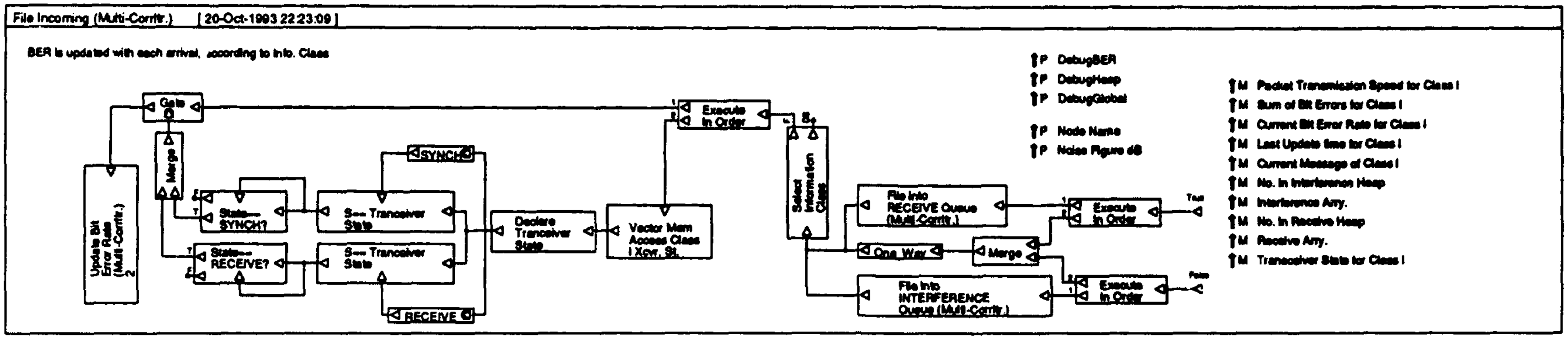
SPSPC Channel Effects [20-Oct-1993 22:13:01]
Input Signal
Select Source
I1 == I2
Name to Address
Select Source Position
Select Source Waypoint
Select Speed
Select center frequency
Select Power dB
Gate
Insert Power dB
Insert Doppler Shift Amount
Propagation Delay
Signal at Receiver
Antenna Gains = 0 dB
Propagation Loss
Rcvr. Loss Factor dB
INFO <-real
Execute In Order
Propgn. Power Loss Switch == ON?
Prop. Pwr. Ls. Swtch.
Relative Velocity (nmi)
Doppler Shift 2
Current Position and Waypoint
Node Speed
Range (nmi)
R/
Speed of Light (nmi/sec)
P Node Name
P DebugChannel
P DebugGlobal
M Node Speed
M Longitude
M Latitude
M Altitude
M Waypoint Longitude
M Waypoint Latitude
M Waypoint Altitude
P Propagation Power Loss Switch
P Receiver Power Loss Factor dB
P Propagation Loss Exponent



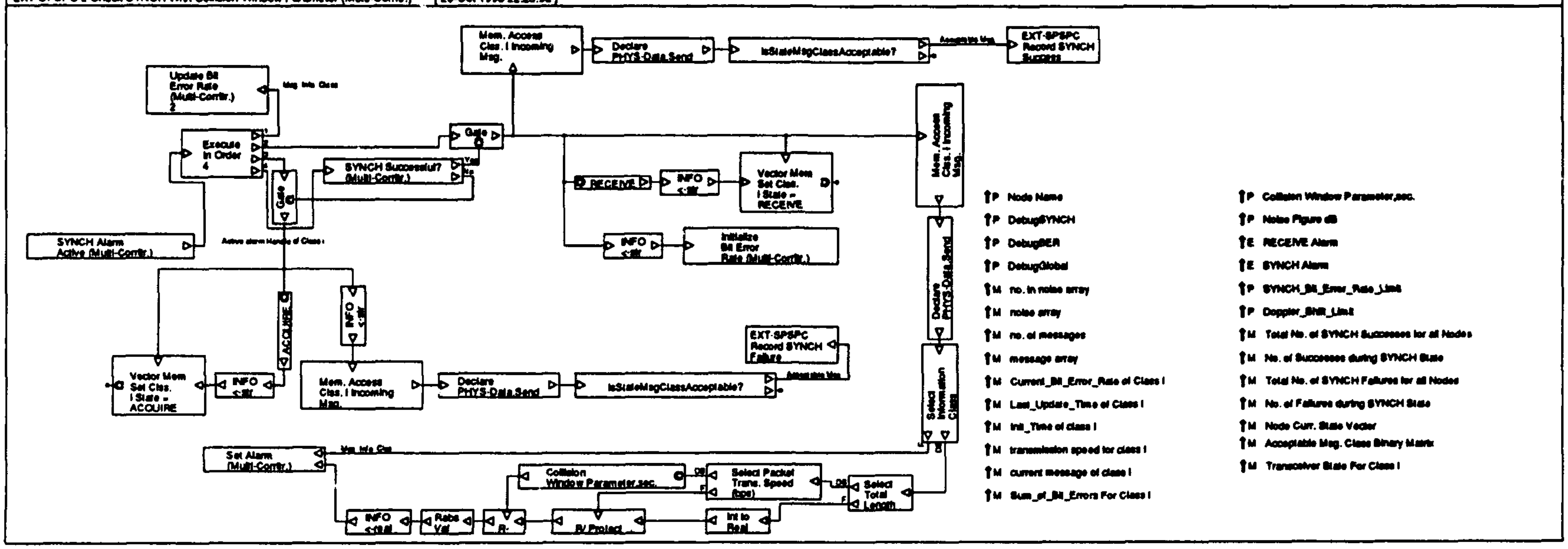
EXT-SPSPC Receiver 5 With Multi Correlators
EXT-SPSPC Record TO TRANSMISSIONS
EXT-SPSPC2 Record TRANSMITTER BUSY LOSSES
SPSPC XverState PktSwitch 3 (Multi-Corrltr.)
EXT-SPSPC2 Receiver SYNCH State Processing (Multi-Corrltr.)
Merge
Execute In Order
INFO
Mem. Access Curr. St. Var.
Name to Address
Should Node Change State For This MsgClass?
Select Destination
Select Information Class
Mem. Read Reception Option
Mem. Read Transmission Option
Mem. Write
Simple FIFO
Gate
Select Absolute Index Number
Select Packet Trans. Speed
This Node Address
Broadcast Address
File Incoming (Multi-Corrltr.)
Node Name
Noise Figure dB
No. of Info. Classes
Transceiver State for Class i
Number of Receiver Busy Collisions
Total No. of COLLISIONS for all Nodes
No. of Collisions during RECEIVE State
No. of Collisions during SYNCH State
Total No. of SYNCH Failures for all Nodes
No. of Failures during SYNCH State
No. of Non-Received Packets
Node Curr. State Vector
Reception Option
Transmission Option
Noise List
Init_Time For Class i
Message array
No. on Receive Heap
Interference Array



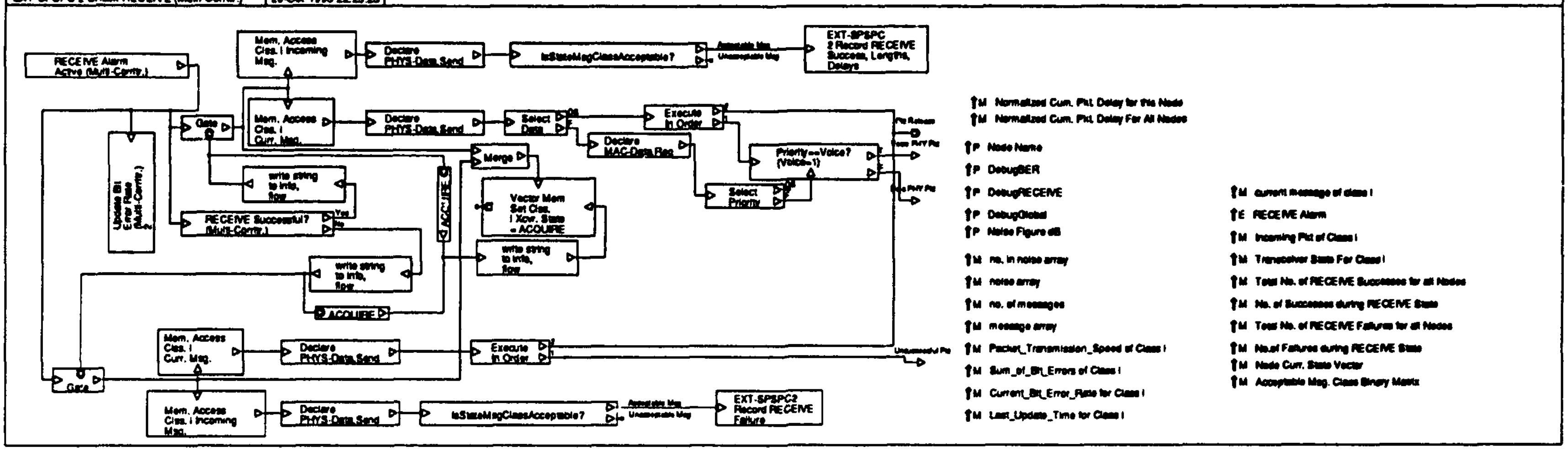
EXT-SPSPC Receiver 5
20-Oct-1993 22:17:40
EXT-SPSPC Record Non-Received Messages
EXT-SPSPC Record TO TRANSMISSIONS
EXT-SPSPC2 Record TRANSMITTER BUSY LOSSES
EXT-SPSPC2 Receiver RECEIVE State Processing
EXT-SPSPC2 Receiver ACQUIRE State Processing
EXT-SPSPC2 Receiver SYNCH State Processing
EXT-SPSPC2 Check RECEIVE
SPSPC XvrStatePktSwitch3
Merge
Execute In Order
Select Destination
Select Information Class
Mem.Access Curr.St.Vctr.
Name to Address
Broadcast Address
This Node Address
Select Packet Trans. Speed
Select Absolute Index Number
Mem.Read Reception Option
Mem.Read Transmission Option
Mem.Write Pkt Xmssn speed
Gate
INFO
Node Name
DebugHeap
DebugRECEIVE
DebugSYNCH
DebugBER
DebugGlobal
Noise Figure dB
Collision Window Parameter
Transceiver State
Reception Option
Transmission Option
packet transmission speed
Message array
Sum of Bit errors
Noise Lvl.
Current_Bit_Error_Rate
Last_Update_Time
Init_Time
SYNCH alarm
RECEIVE alarm
current message
No. on Receive Heap
No. in Interference Heap
Interference Array
Node Curr. State Vector
Node Msg. Redirected State Matrix
Transmission State-Msg.Class Probability Array


File Incoming (Multi-Contr.)
[20-Oct-1993 22:23:09]
BER is updated with each arrival, according to Info. Class
Gate
Update Bit Error Rate (Multi-Contr.)
Merge
State---SYNCH?
State---RECEIVE?
S--- Tranceiver State
S--- Tranceiver State
SYNCH
RECEIVE
Declare Tranceiver State
Vector Mem Access Class i Xcvr. St.
Execute In Order
Select Information Class
File into RECEIVE Queue (Multi-Contr.)
One Way
Merge
File into INTERFERENCE Queue (Multi-Contr.)
Execute In Order
Execute In Order
True
False
P DebugBER
P DebugHeap
P DebugGlobal
P Node Name
P Noise Figure dB
M Packet Transmission Speed for Class i
M Sum of Bit Errors for Class i
M Current Bit Error Rate for Class i
M Last Update time for Class i
M Current Message of Class i
M No. in Interference Heap
M Interference Arry.
M No. in Receive Heap
M Receive Arry.
M Tranceiver State for Class i

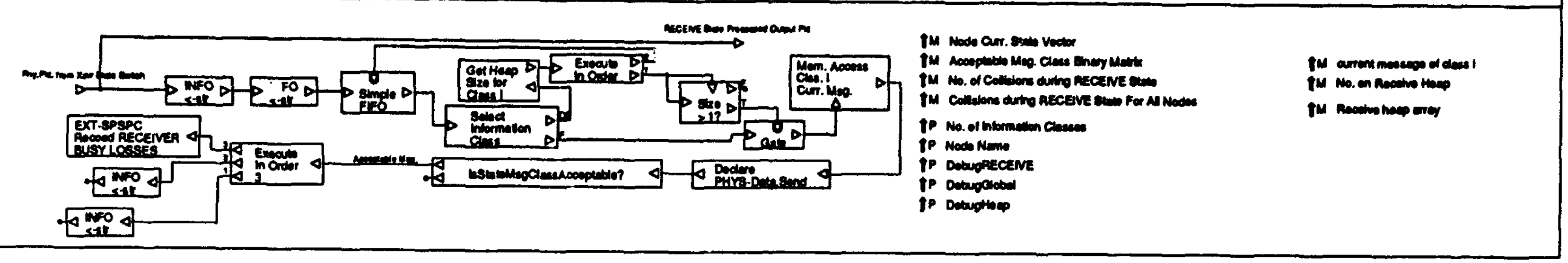
EXT-SPSPC 2 Check SYNCH With Collision Window Parameter (Multi-Comfr.) [20-Oct-1993 22:23:52]



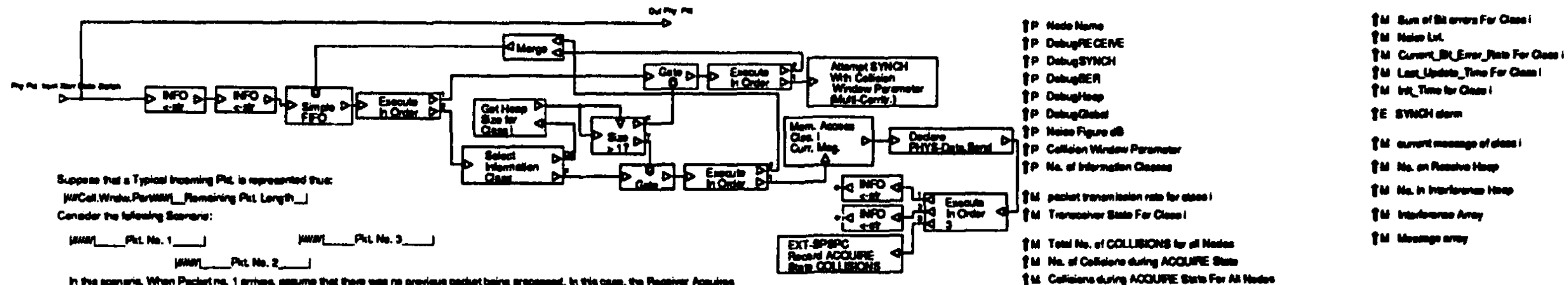
EXT-SPSPC 2 Check RECEIVE (Multi-Carrier) [20-Oct-1993 22:25:23]



EXT-SPSPC 2 Receiver RECEIVE State Processing (Multi-Contr.) [20-Oct-1993 22:26:09]



EXT-SPSPC 2 Receiver ACQUIRE State Proc. with Coll. Wndw. Param. (Multi-Contr.) [20-Oct-1993 22:28:51]



Suppose that a Typical Incoming Pkt. is represented thus:

|##Coll.Wndw.Par###|__Remaining Pkt. Length__|

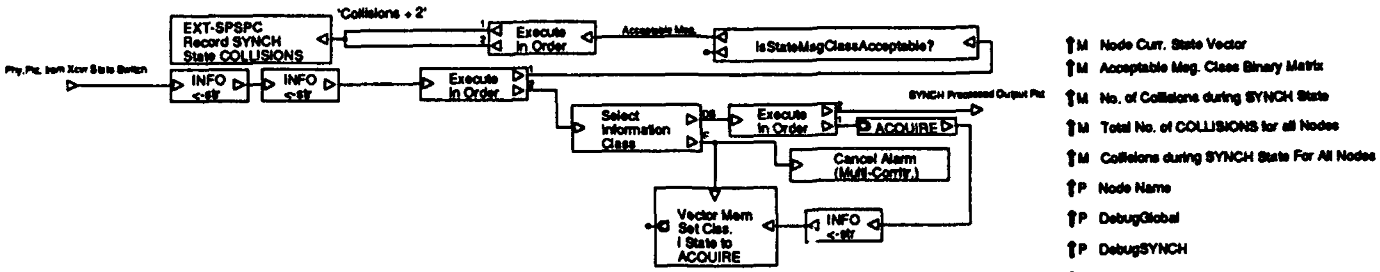
Consider the following Scenario:

|####|____Pkt. No. 1____| |####|____Pkt. No. 3____|

|####|____Pkt. No. 2____|

In this scenario, When Packet no. 1 arrives, assume that there was no previous packet being processed. In this case, the Receiver Acquires it and then (provided no collision occured during the Acquire State) attempts to Synchronize with it and finally, (provided no collision occurred during Synchronization) proceeds to Receive it. After processing the last bit of Pkt. No.1 the Receiver reverts to its Acquire State. Hence, the Receiver's Heap size which was 0 prior to Pkt No. 1's arrival is now 1 and remains 1 for the duration of reception of Pkt. No. 1's full length. If Pkt. No. 2 arrives, but after, i.e., outside of Pkt. No. 1's Collision Window, it encounters the Receiver in the Receive State and is considered as 'lost due to a Busy receiver'. Even though Pkt. No. 2 is considered lost, the dynamics of the communication processing cause the Receiver (which has successfully completed receiving Pkt. no. 1) to switch back to the Acquire State while still being occupied in processing the remnants of Pkt. No. 2, and as mentioned above, its Heap size is still 1. Thus, when Pkt. 3 arrives 'on the heels' of Pkt. No. 2 the heap size is bumped up to 2 and it encounters the Receiver in the Acquire State but with a Heap size > 1, which implies an Acquire State collision incurred by Pkt. No. 3.

EXT-SPSPC2 Receiver SYNCH State Processing (Multi-Corrltr.) [20-Oct-1993 22:27:56]



Suppose that a Typical Incoming Pkt. is represented thus:

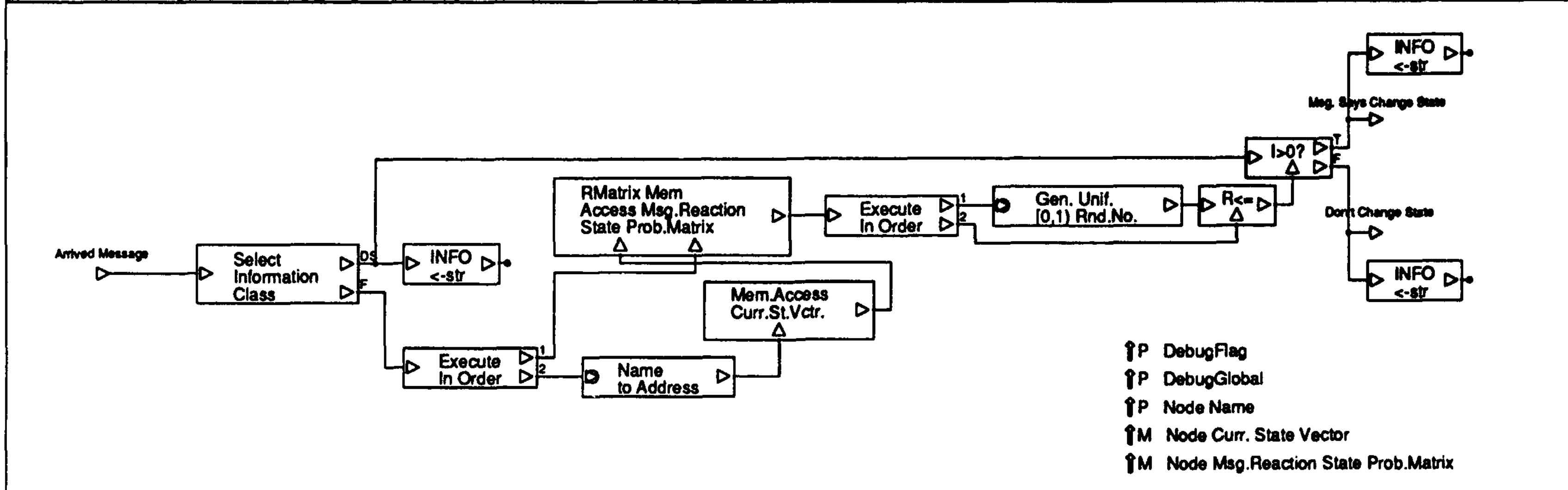
|///Coll.Wndw.Part/////|__Remaining Pkt. Length__|

Consider the following Scenario (which differs from the scenario in Module ACQUIRE State Processing with Collision Window):

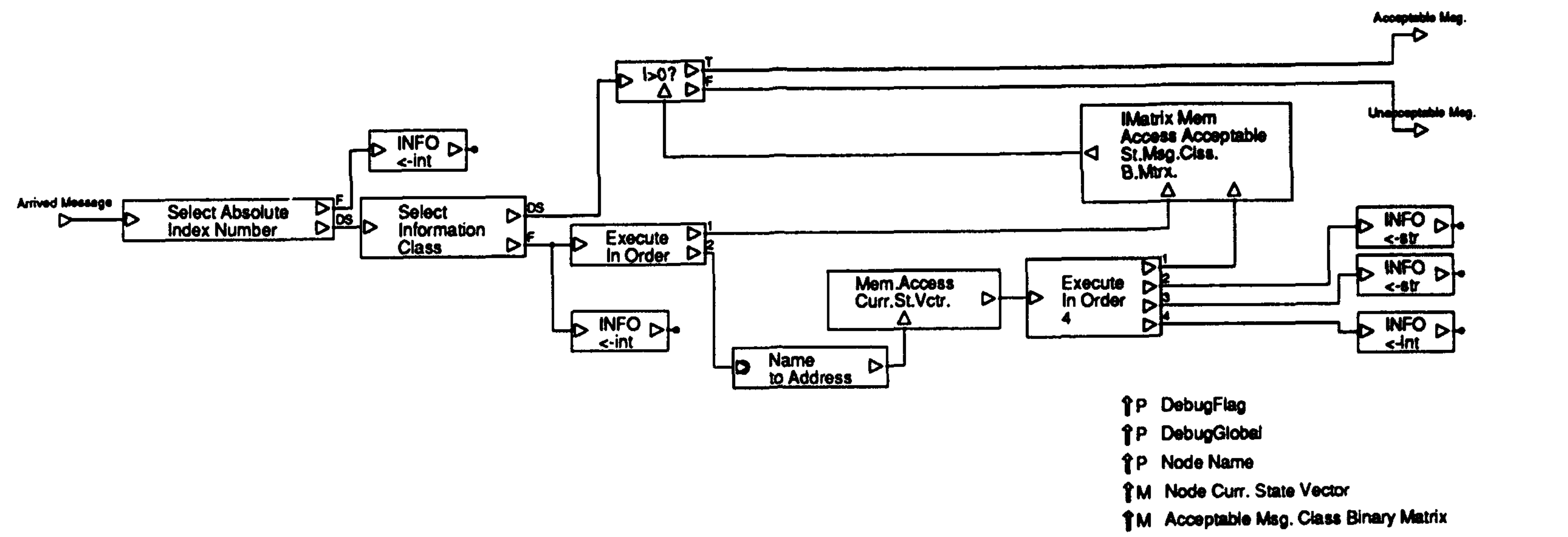
|//////|_____Pkt. No. 1_______| |//////|______Pkt. No. 2________|
|//////|____________Pkt. No. 2____________|

In this scenario, When Packet no. 1 arrives, assume that there was no previous packet being processed. In this case, the Receiver Acquires it and then (provided no collision occured during the Acquire State) attempts to Synchronize with it and finally, (provided no collision occurred during Synchronization) proceeds to Receive it. After processing the last bit of Pkt. No.1 the Receiver reverts to its Acquire State. Hence, the Receiver's Heap size which was 0 prior to Pkt No. 1's arrival is now 1 and remains 1 for the duration of reception of Pkt. No. 1's full length. If Pkt. No. 2 arrives, but WITHIN Pkt. No. 1's Collision Window, it encounters the Receiver in the SYNCH State and BOTH, Pkt. Nos. 1 and 2 are considered as 'collisions in the SYNCH State'. Even though Pkt. No. 2 is considered lost (as is Pkt. No. 1), the dynamics of the communication processing cause the Receiver (which has unsuccessfully completed receiving Pkt. no. 1) to switch back to the Acquire State while still being occupied in processing the remnants of Pkt. No. 2, and as mentioned above, its Heap size is still 1. Thus, when Pkt. no. 3 arrives 'on the heels' of Pkt. No. 2 the heap size is bumped up to 2 and it encounters the Receiver in the Acquire State but with a Heap size > 1, which implies an Acquire State collision incurred by Pkt. No. 3.

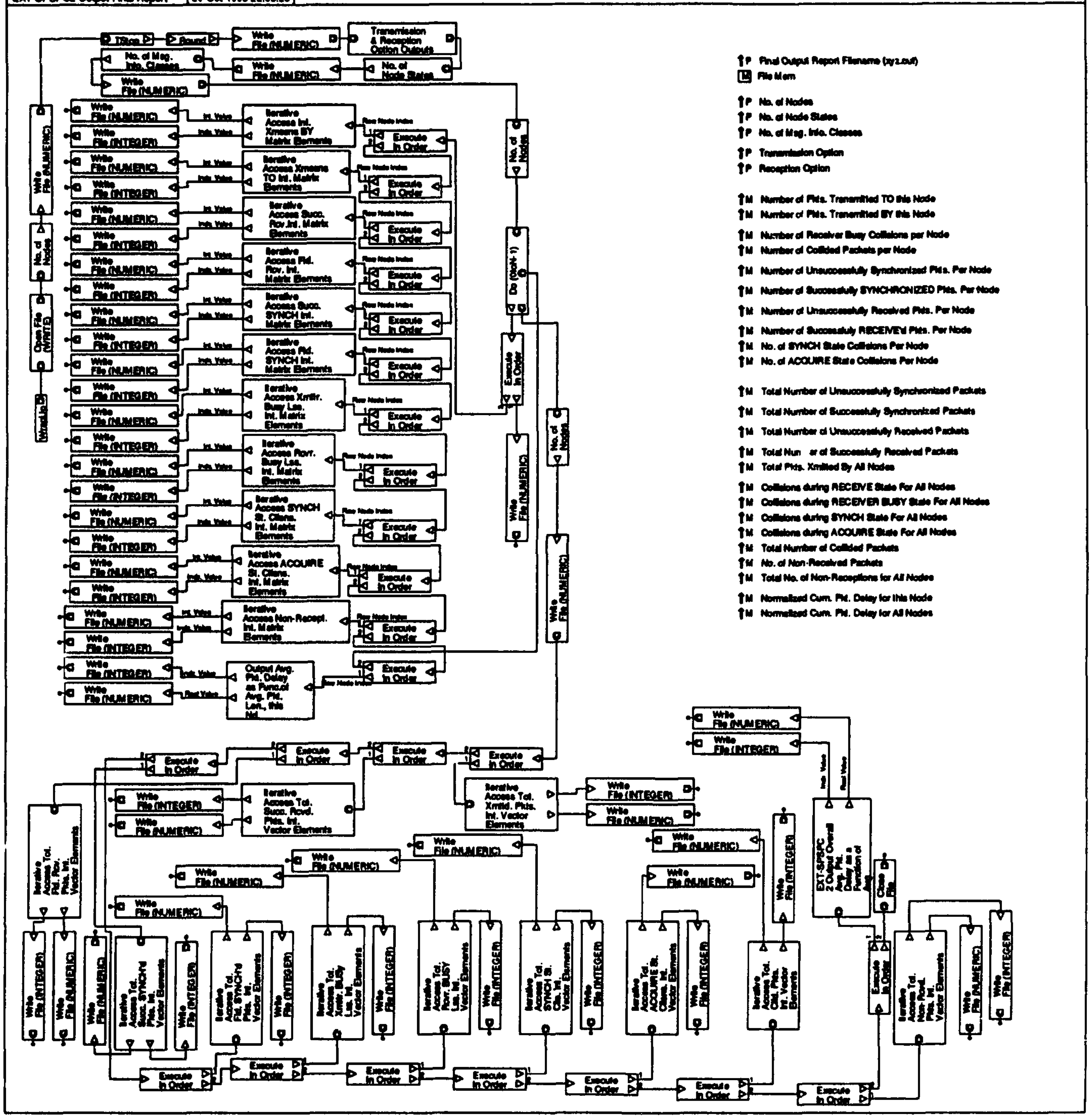
Should Node Change State For This MsgClass? [20-Oct-1993 22:29:05]



isStateMsgClassAcceptable? [20-Oct-1993 22:29:33]



EXT-SPSPC2 Output Final Report [20-Oct-1993 22:03:23]



Name: MAC-Data.Req [802.x Data Structure]
Date: Wednesday, 10/20/93 10:32:31 pm PST

Name	Type	Subrange	Default Value
Source	INTEGER	(-Infinity, +Infinity)	...
Destination	INTEGER	(-Infinity, +Infinity)	...
Data	ROOT-OBJECT	...	...
Length	INTEGER	[0, +Infinity)	...
Priority	INTEGER	[0, +Infinity)	0
Service Class	MAC-Service Class	...	Asynchronous

Name: PHYS-Data.Send [LPI-WORKING]
Date: Wednesday, 10/20/93 10:30:38 pm PST

Name	Type	Subrange	Default Value
Source	INTEGER	(-Infinity, +Infinity)	0
Destination	INTEGER	(-Infinity, +Infinity)	0
Data	ROOT-OBJECT	...	...
Total Length	INTEGER	[0, +Infinity)	0
Priority	INTEGER	[0, +Infinity)	0
Source Position	Position	...	...
Source Waypoint	Position	...	...
Source Speed	REAL	[0, +Infinity)	0.0
Power	REAL	(-Infinity, +Infinity)	0.0
Number of Sibling Packets	INTEGER	(0, +Infinity)	...
Antenna Parameters	Antenna Parameters	...	...
Antenna Type	Antenna Type	...	Isotropic
Time Generated	REAL	[0, +Infinity)	0.0
code rate	REAL	[0, 1]	0.5
Power Control	INT-VECTOR	...	...
Absolute Index Number	INTEGER	[0, +Infinity)	0
Doppler Shift Amount	REAL	(-Infinity, +Infinity)	0.0
Preamble Length	INTEGER	[0, +Infinity)	0
Modulation Type	Mod Type	...	BPSK
Total Time	REAL	(-Infinity, +Infinity)	0.0
center frequency	REAL	[0, +Infinity)	0.0
hopping bandwidth	REAL	[0, +Infinity)	0.0
number of bins	INTEGER	[0, +Infinity)	0
direct sequence bandwidth	REAL	[0, +Infinity)	0.0
Packet Trans. Speed (bps)	REAL	[0, +Infinity)	0.0
Information Class	INTEGER	[0, +Infinity)	0
Sibling Packet Index	INTEGER	(0, +Infinity)	...
Source MBMR Address	INT-VECTOR	...	...
Destination MBMR Address	INT-VECTOR	...	...
Next Node MBMR Address	INT-VECTOR	...	...
MBMR Channel ID	INTEGER	[0, +Infinity)	...